FLOW CONTROL TECHNIQUES FOR REAL-TIME MEDIA APPLICATIONS IN BEST-EFFORT NETWORKS USING FLUID MODELS

A Thesis

by

APOSTOLOS KONSTANTINOU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2004

Major Subject: Mechanical Engineering

FLOW CONTROL TECHNIQUES FOR REAL-TIME MEDIA APPLICATIONS IN BEST-EFFORT NETWORKS USING FLUID MODELS

A Thesis

by

APOSTOLOS KONSTANTINOU

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Alexander Parlos (Chair of Committee)	Suhada Jayasuriya (Member)
Jennifer Welch (Member)	Dennis O'Neal (Interim Head of Department)

August 2004

Major Subject: Mechanical Engineering

ABSTRACT

Flow Control Techniques for Real-Time Media Applications in Best-Effort Networks Using Fluid Models. (August 2004)

Apostolos Konstantinou, B.Eng., National Technical University of Athens

Chair of Advisory Committee: Dr. Alexander Parlos

Quality of Service (QoS) in real-time media applications is an area of current interest because of the increasing demand for audio/video, and generally multimedia applications, over best effort networks, such as the Internet. Media applications are transported using the User Datagram Protocol (UDP) and tend to use a disproportionate amount of network bandwidth as they do not perform congestion or flow control. Methods for application QoS control are desirable to enable users to perceive a consistent media quality. This can be accomplished by either modifying current protocols at the transport layer or by implementing new control algorithms at the application layer irrespective of the protocol used at the transport layer.

The objective of this research is to improve the QoS delivered to end-users in real-time applications transported over best-effort packet-switched networks. This is accomplished using UDP at the transport layer, along with adaptive predictive and reactive control at the application layer. An end-to-end fluid model is used, including the source buffer, the network and the destination buffer. Traditional control techniques, along with more advanced adaptive predictive control methods, are considered in order to provide the desirable QoS and make a best-effort network an attractive channel for interactive multimedia applications. The effectiveness of the control methods, is examined using a Simulink-based fluid-level simulator in combination with trace files extracted from the well-known network simulator ns-2. The

results show that improvement in real-time applications transported over best-effort networks using unreliable transport protocols, such as UDP, is feasible. The improvement in QoS is reflected in the reduction of flow loss at the expense of flow dead-time increase or playback disruptions or both.

To my grandparents

ACKNOWLEDGMENTS

I would like to thank Dr. Alexander Parlos, for giving me the opportunity to work under him. I would like to thank him for his friendly and encouraging attitude and the support he provided. He has been a good teacher and a good advisor. I also owe my thanks to Dr. Suhada Jayasuriya and Dr. Jennifer Welch for their time. Thanks are due to my friends Aninda, Parasuram, Srikar and Vivek. Lastly, I would like to thank my parents and brother. Without their support, I would have never come this far.

TABLE OF CONTENTS

CHAPTER Page

I INTRODUCTION 1

A. Motivation and Objectives 1
B. Literature Review 3
1. Review of Network Related Literature 3
2. Review of Control Related Literature 5
C. Proposed Solution 7
D. Contributions of This Work 9
E. Thesis Overview 10

II COMPLEXITIES OF END-TO-END FLOW CONTROL IN DISTRIBUTED REAL-TIME APPLICATIONS 11

A. Issues Regarding the End-to-End System 11
1. Fluid Level vs Packet Level Simulation 11
2. Unreliable Transport Protocol, Ordered Flows and Packet Losses 12
3. The Time-Varying Nature of the End-to-End Time-Delay 14
4. The Network Topology 15
B. The End-to-End Control Problem 17
1. Complexity of the Control Problem 17
2. Proposed System Configuration 21
C. Assumptions 23
D. Chapter Summary 24

III DYNAMIC MODELING OF THE END-TO-END FLOW TRANSPORT SYSTEM 25

A. Source Buffer Model 25
B. Network Dynamic Model 27
1. Time-Varying Time Delay Model 27
2. Congested Network Router - Output Flow Splitting - Loss Model 30
C. Destination Buffer Model 34
D. Software Used for Simulation Studies 36

CHAPTER Page

E. Chapter Summary 36

IV CONTROLLER DEVELOPMENT 37

A. Desired Controller Performance Criteria - Controller Objectives 37
B. Available Control Strategies 38
1. Reactive Feedback Compensation 39
2. Predictive Control Laws 39
3. Linear and Nonlinear Control Laws 40
C. Controller Development 40
1. Linear Controller (Controller 1) 40
2. Nonlinear Compensation Based on End-to-End Delay Measurement Feedback. (Controller 2) 41
a. Reactive Nonlinear Compensation Based on End-to-End Delay 41
b. Predictive Nonlinear Compensation Based on End-to-End Delay 43
3. Nonlinear Compensation Based on Accumulation Measurement Feedback (Controller 3) 44
a. Reactive Nonlinear Compensation Based on Accumulation 44
b. Predictive Nonlinear Compensation Based on Accumulation and Delay 46
4. Compensation Based on Model Predictive Control Techniques (Controller 4) 46
D. Chapter Summary 51

V SIMULATION RESULTS 52

A. Network Architecture 52
B. Effectiveness of the Controllers for Baseline Application Send Rate 54
1. Open-Loop Simulations 54
2. Linear Control - Controller 1 65
3. Nonlinear Control - Controller 2 70
a. Reactive Implementation of Controller 2 70
b. Predictive Implementation of Controller 2 75
c. The Effect of the Controller Gain k_2 77
4. Nonlinear Control - Controller 3 81

CHAPTER Page

a. Reactive Implementation of Controller 3 81
b. Predictive Implementation of Controller 3 88
5. Model Predictive Control - Controller 4 91
6. Importance of the Initial Source Buffering 110
C. Effectiveness of Flow Controllers for 20% Decrease in Application Send Rate 117
1. Open-Loop Simulation 117
2. Reactive Implementation of Controller 2 117
3. Predictive Implementation of Controller 2 117
4. Reactive Implementation of Controller 3 123
5. Predictive Implementation of Controller 3 128
D. Effectiveness of Flow Controllers for 20% Increase in Application Send Rate 131
1. Open-Loop Simulation 131
2. Reactive Implementation of Controller 2 131
3. Predictive Implementation of Controller 2 131
4. Reactive Implementation of Controller 3 141
5. Predictive Implementation of Controller 3 144
E. Comparison of Controller Performance 149
F. Chapter Summary 156

VI SUMMARY AND CONCLUSIONS 158

A. Summary 158
B. Conclusions 159
C. Recommendations for Future Work 160

REFERENCES 162

VITA 169

LIST OF TABLES

TABLE		Page
I	Performance Parameters for the Open-Loop Case Using Different Cross-Traffic Traces.	65
II	20-Step-Ahead Prediction Errors	97
III	A Comparison of Controller Performance for Baseline Send Rate Using Cross-Traffic 1 Trace.	150
IV	A Comparison of Controller Performance for Baseline Send Rate Using Cross-Traffic 2 Trace.	151
V	A Comparison of Controller Performance for 120 ups Send Rate Using Cross-Traffic 1 Trace.	152
VI	A Comparison of Controller Performance for 180 ups Send Rate Using Cross-Traffic 1 Trace.	153
VII	An Overall Comparison of Controller Performance.	155

LIST OF FIGURES

FIGURE		Page
1	A Block Diagram of the Proposed System.	8
2	End-to-End System Showing the Send Rate, Arrival Rate and Packet Loss Rate of a Single Flow.	14
3	Simplified Network Topology.	16
4	(a) End-to-End System with No Control and (b) System with Application Layer Feedback.	18
5	A Block Diagram of the Proposed Control System Using Feedback of Delay, Accumulation and Buffer Level.	22
6	Block Diagram of the System Showing a Source Buffer.	26
7	A Block Diagram of the End-to-End System with Model Predictive Control.	47
8	Cross-Traffic Traces Used in MATLAB Simulations	53
9	Buffer Level and Flow Rates for Open-Loop Simulation Using Cross-Traffic 1.	55
10	End-to-End Delay and Losses for Open-Loop Simulation Using Cross-Traffic 1.	56
11	Cumulative Flow Rates and Accumulation for Open-Loop Simulation Using Cross-Traffic 1.	57
12	Buffer Level and Flow Rates for Open-Loop Simulation Using Cross-Traffic 1 for Increased x_{on}.	59
13	End-to-End Delay and Losses for Open-Loop Simulation Using Cross-Traffic 1 for Increased x_{on}.	60

FIGURE Page

14 Cumulative Flow Rates and Accumulation for Open-Loop Simulation Using Cross-Traffic 1 for Increased x_{on}. 61

15 Buffer Level and Flow Rates for Open-Loop Simulation Using Cross-Traffic 2 for Increased x_{on}. 62

16 End-to-End Delay and Losses for Open-Loop Simulation Using Cross-Traffic 2 for Increased x_{on}. 63

17 Cumulative Flow Rates and Accumulation for Open-Loop Simulation Using Cross-Traffic 2 for Increased x_{on}. 64

18 Buffer Level and Flow Rates for Controller 1 Simulation Using Cross-Traffic 1. 66

19 End-to-End Delay, Source Buffer Level and Losses for Controller 1 Simulation Using Cross-Traffic 1. 67

20 Buffer Level and Flow Rates for Controller 1 Simulation Using Cross-Traffic 2. 68

21 End-to-End Delay, Source Buffer Level and Losses for Controller 1 Simulation Using Cross-Traffic 2. 69

22 Buffer Level and Flow Rates for Reactive Controller 2 Simulation Using Cross-Traffic 1. 71

23 End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 2 Simulation Using Cross-Traffic 1. 72

24 Buffer Level and Flow Rates for Reactive Controller 2 Simulation Using Cross-Traffic 2. 73

25 End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 2 Simulation Using Cross-Traffic 2. 74

26 Buffer Level and Flow Rates for Predictive Controller 2 Simulation Using Cross-Traffic 1. 75

27 End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 2 Simulation Using Cross-Traffic 1. 76

FIGURE Page

28 Buffer Level and Flow Rates for Predictive Controller 2 Simulation Using Cross-Traffic 2. 77

29 End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 2 Simulation Using Cross-Traffic 2. 78

30 Buffer Level and Flow Rates for Predictive Controller 2 Simulation Using Cross-Traffic 1 for $k_2 = 0$. 79

31 End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 2 Simulation Using Cross-Traffic 1 for $k_2 = 0$. 80

32 Buffer Level and Flow Rates for Reactive Controller 3 Simulation Using Cross-Traffic 1. 82

33 End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 3 Simulation Using Cross-Traffic 1. 83

34 Cumulative Flow Rates and Accumulation for Reactive Controller 3 Simulation Using Cross-Traffic 1. 84

35 Buffer Level and Flow Rates for Reactive Controller 3 Simulation Using Cross-Traffic 2. 85

36 End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 3 Simulation Using Cross-Traffic 2. 86

37 Cumulative Flow Rates and Accumulation for Reactive Controller 3 Simulation Using Cross-Traffic 2. 87

38 Buffer Level and Flow Rates for Predictive Controller 3 Simulation Using Cross-Traffic 1. 88

39 End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 3 Simulation Using Cross-Traffic 1. 89

40 Cumulative Flow Rates and Accumulation for Predictive Controller 3 Simulation Using Cross-Traffic 1. 90

41 End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 3 Simulation Using Cross-Traffic 2. 91

FIGURE Page

42 End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 3 Simulation Using Cross-Traffic 2. 92

43 Cumulative Flow Rates and Accumulation for Predictive Controller 3 Simulation Using Cross-Traffic 2. 93

44 Send Rate Used to Identify the Open-Loop System Model. 94

45 20-Step-Ahead Prediction of Destination Buffer Level Using Cross-Traffic 1 for Application Send Rate of 150 ups. 95

46 20-Step-Ahead Prediction of Destination Buffer Level Using Cross-Traffic 2 for Application Send Rate of 150 ups. 96

47 20-Step-Ahead Prediction of Destination Buffer Level Using Cross-Traffic 1 for Application Send Rate of 120 ups. 98

48 20-Step-Ahead Prediction of Destination Buffer Level Using Cross-Traffic 1 for Application Send Rate of 180 ups. 99

49 Buffer Level and Flow Rates for MPC Simulation Using Cross-Traffic 1; Destination Buffer Regulation. 101

50 End-to-End Delay, Source Buffer Level and Losses for MPC Simulation Using Cross-Traffic 1; Destination Buffer Regulation. 102

51 Buffer Level and Flow Rates for MPC Simulation Using Cross-Traffic 2; Destination Buffer Regulation. 103

52 End-to-End Delay, Source Buffer Level and Losses for MPC Simulation Using Cross-Traffic 2; Destination Buffer Regulation. 104

53 Buffer Level and Flow Rates for MPC Simulation Using Cross-Traffic 1; Cumulative Flow Difference Regulation. 106

54 End-to-End Delay, Source Buffer Level and Losses for MPC Simulation Using Cross-Traffic 1; Cumulative Flow Difference Regulation. 107

55 Cumulative Flow Rates and Accumulation for MPC Simulation Using Cross-Traffic 1; Cumulative Flow Difference Regulation. 108

FIGURE Page

56 20-Step-Ahead Prediction of Cumulative Flow Difference between Sending and Arrival Rate Using Cross-Traffic 1. 109

57 Buffer Level and Flow Rates for MPC Simulation Using Cross-Traffic 2; Destination Buffer Regulation. 111

58 End-to-End Delay, Source Buffer Level and Losses for MPC Simulation Using Cross-Traffic 2; Destination Buffer Regulation. 112

59 Buffer Level and Flow Rates for MPC Simulation Using Cross-Traffic 2 for 30% Decreased Initial Source Buffering; Destination Buffer Regulation. 113

60 End-to-End Delay, Source Buffer Level and Losses for MPC Simulation Using Cross-Traffic 2 for 30% Decreased Initial Source Buffering; Destination Buffer Regulation. 114

61 Buffer Level and Flow Rates for MPC Simulation Using Cross-Traffic 2 for 65% Decreased Initial Source Buffering; Destination Buffer Regulation. 115

62 End-to-End Delay, Source Buffer Level and Losses for MPC Simulation Using Cross-Traffic 2 for 65% Decreased Initial Source Buffering; Destination Buffer Regulation. 116

63 Buffer Level and Flow Rates for Open-Loop Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups. 118

64 End-to-End Delay and Losses for Open-Loop Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups. 119

65 Cumulative Flow Rates and Accumulation for Open-Loop Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups. 120

66 Buffer Level and Flow Rates for Reactive Controller 2 Using Cross-Traffic 1 for Application Send Rate of 120 ups. 121

67 End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups. 122

FIGURE Page

68 Buffer Level and Flow Rates for Predictive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups. 123

69 End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups. 124

70 Buffer Level and Flow Rates for Reactive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups. 125

71 End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups. 126

72 Cumulative Flow Rates and Accumulation for Reactive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups. 127

73 Buffer Level and Flow Rates for Cross-Traffic 1 for Predictive Controller 3 Simulation Using Application Send Rate of 120 ups. . . 128

74 End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups. 129

75 Cumulative Flow Rates and Accumulation for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups. 130

76 Buffer Level and Flow Rates for Open-Loop Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups. 132

77 End-to-End Delay and Losses for Open-Loop Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups. 133

78 Cumulative Flow Rates and Accumulation for Open-Loop Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups. 134

79 Buffer Level and Flow Rates for Reactive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups. 135

FIGURE Page

80 End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups. 136

81 Buffer Level and Flow Rates for Predictive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups. 137

82 End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups. 138

83 Buffer Level and Flow Rates for Predictive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups and $k_2 = 0$. 139

84 End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups and $k_2 = 0$. 140

85 Buffer Level and Flow Rates for Reactive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups. 141

86 End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups. 142

87 Cumulative Flow Rates and Accumulation for Reactive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups. 143

88 Buffer Level and Flow Rates for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups. 144

89 End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups. 145

90 Cumulative Flow Rates and Accumulation for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups. 146

FIGURE Page

91 Buffer Level and Flow Rates for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups and $k_2 = 0.2$. 147

92 End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups and $k_2 = 0.2$. 148

CHAPTER I

INTRODUCTION

This thesis presents a solution to some of the problems encountered when transporting live interactive media applications, e.g. video, over best-effort networks, e.g. the Internet. Best-effort networks are characterized by variations in end-to-end delay, available bandwidth, and packet losses and create challenges for packet transport of interactive applications.

A system theoretic framework is developed for modeling and media application control over best-effort networks using unreliable transport protocols. Fluid-based models are used for designing and testing the developed controllers.

A. Motivation and Objectives

The last several years have seen a significantly increasing demand for audio/video and general multimedia applications over best effort networks. Half-duplex applications, both stored and live streamed media, are widely used and have created new trends in network communication. In this research the case of real-time multimedia is investigated.

Real-time applications are more time sensitive than non-real-time applications which can be buffered extensively at the destination side. Although the Transport Control Protocol (TCP) has become the default standard for data transmission control on the Internet, because of the flow and congestion control it performs, most media applications use User Datagram Protocol (UDP) as their transport protocol. Congestion control aids in network health but not in the QoS delivered to the end-

---

The journal model is *IEEE Transactions on Automatic Control.*

users, as QoS highly depends upon keeping the sending rate above the minimum rate required for the media while minimizing packet losses. Congestion control causes reduction of the sending rate following the onset of transient congestion which in turn causes packets to arrive too late to their destination. Consequently, destination buffers empty and playback is disrupted.

On the other hand, the growth of unresponsive flows on the Internet could result in increased congestion which in turn could cause packet losses and playback disruptions due to lost information. Therefore, as non-"TCP-friendly" traffic increases, it is more difficult for quality (QoS) guarantees to be provided. The goal of this research is to show that regulating the media flow rate at the application layer of the sender using closed-loop feedback control can improve the QoS delivered to the receivers. The proposed control techniques are implemented at the application layer on top of an unreliable transport protocol, such as UDP, and flows are transported over a best-effort wide area network.

In order to address this specific problem, a model of the end-to-end system under some simplifying assumptions is developed. An end-to-end transport model is the system of interest, where end-to-end describes system variables from the source (sender) to the destination (receiver). In this end-to-end system a simplified network model is included. The flow to be controlled is assumed to travel one and the same end-to-end path, as the network is modelled as a single congested router.

QoS is a user-perceived measure of performance. When a video or audio stream is transmitted from the sender over a best effort network, packets might be lost or delayed. As a result the playback at the destination is displayed with reduced quality and the user perceives poor service. Feedback signals from the destination are used to regulate the stream rate at the source in an attempt to impact the destination playback and improve the quality delivered to the receiver, for real-time applications.

Improving the QoS delivered by real-time, media applications can be achieved by establishing the following performance objectives:

- Minimizing the playback start time. The time elapsed between the start of the media flow at the source application and the start of playback at the destination should be kept as small as possible. This is called the system dead-time.
- Minimize packet losses to guarantee complete information transfer to the end-user. A positive side-effect of this objective is that if it is met, application throughput is increased and therefore network bandwidth is used in a more efficient way.
- Prevent a disruption from occurring during media playback to maintain the QoS perceived by the end-user. This is directly related to network jitter and it can be accomplished by adapting to the variations in end-to-end delays experienced by the application. It is typically accomplished by buffering on the receiver side.

B. Literature Review

The Literature related to this thesis is separated into two categories: Literature related to computer networks and literature related to controls.

1. Review of Network Related Literature

The deployment of multi-cast and real-time audio/video streaming and other interactive multimedia applications has increased the percentage of non-"TCP-friendly" traffic on the Internet. These applications are time sensitive and they do not employ congestion control [1, 2]. Applications built on TCP, such as Web browsers, email

clients and FTP, share the bandwidth fairly while applications using unresponsive protocols, such as UDP, compete unfairly with TCP traffic, sometimes to the potential of congestion collapse. Floyd [3] talks about all the negative impacts of the increased deployment of non-congestion-controlled best-effort traffic. Attempts have been made to create new hybrid protocols incorporating aspects from both TCP and UDP [4]. In any case, any new transport protocol developed, must be compared to TCP to make sure it does not adversely affect existing TCP traffic. Several models of TCP are available for this purpose [5]. Generally, efficient management of network traffic, as well as methods for traffic characterization have been extensive research areas [6, 7].

Control problems associated with distributed real-time applications are complex and modeling should be the first step before attempting to address them. However modeling and simulation of a network such as the Internet is not an easy task [8]. In [9] the use of fluid simulation techniques is discussed, as well as its advantages over packet level simulation. As today's networks grow heterogeneously, traditional packet level simulations have been inefficient in capturing and accurately evaluating overall network performance. Fluid-based simulations have been proposed to cope with the problem. Fluid-based methodologies enable the solution of large networks handling large numbers of responsive flows (e.g., TCP-based) and non-responsive flows (e.g., UDP-based). In [10, 11, 12] some fluid techniques for solving a network of routers are developed, active queue management policies (e.g., RED) are implemented and techniques for approximating the transient behavior of responsive and non-responsive flows in a network are presented, respectively. In [13], Black et al. model a network using fluid models called "infopipes". The authors compare buffers with tanks, network links to pipes and pumps to control sources. The paper shows the need to apprehend and model the complicated structure of computer networks and the Internet.

Congestion control is an endless area of research [14]. M. Jain and C. Dovrolis [15], talk about the importance of the available bandwidth (avail-bw) in congestion control. They describe a methodology called SLoPS for measuring avail-bw and investigate the relationship with TCP throughput. Imer et al. [16] present a congestion control algorithm for ATM networks.

Predictive control schemes can also be used in developing algorithms for congestion control [17, 18]. Packet loss prediction as well as prediction of round trip times are used.

Interesting is the approach presented in [19, 20]. The authors have used the notion of accumulation, for developing congestion control algorithms. In the present research we attempt to use accumulation as a feedback signal. Accumulation and end-to-end delay both reflect network dynamics, but end-to-end delay is a variable easier to measure in simulations while much more challenging to measure in real-time applications that include packet losses. Accumulation is defined as the difference between the cumulative sending and arrival flow.

Various techniques for the control of packet-based best effort audio/video transmission have been developed using unreliable network protocols [21, 22, 23]. In [24, 25] the authors describe the feedback control of the send rate based on changing video encoder parameters. These papers are concerned with rate control as a means of meeting the requirements of the available network bandwidth resources.

General sources on network and media performance issues are the books by Stallings [26], Minoli and Schmidt [27] and Garcia and Widjaja [28].

2. Review of Control Related Literature

Mangan in [29] attempted the problem of improving the QoS of half-duplex, prerecorded multimedia transmission. The author used several control techniques both

reactive and predictive. He used discrete and continuous-time models based on time-varying transport delays. However, parts of the system used were not modelled as he took a black box approach [30].

Classical and modern control theory have been applied to computer networks. Mascolo [31] proposes classical control theory and Smith's principle as tools for designing effective and simple congestion control algorithms for data networks. The author is concerned about stability of the derived closed-loop control laws.

Active queue management (AQM) is by definition a feedback control mechanism. In [32, 33] control laws based on linearized models of TCP are derived. Classical linear control is applied and compared to existing AQM policies.

Tuning of controllers has always been a challenging problem whether applied to computer networks or other mechanical systems. Different methods for tuning feedback controllers in integrating processes have been investigated [34, 35]. These papers are mainly concerned with system stability in the presence of constant delays without taking into account the effects of time-varying time delays. In [36] optimal control schemes have been applied to time-varying time delayed systems although strict conditions are placed on the variation of the time delays.

Some simplistic fluid-based control techniques are presented in [37]. However, the models used are unrealistically simple and they do not reflect the actual dynamics of a network.

Control and stability of computer networks are addressed in [38, 39] where the network which is the system plant, is modelled as an integrator.

Control in the presence of time-varying time delays has been proposed for use in computer networks as well as in the process industries [36, 40, 41]. The goal of the work of Ataslar et al. [36, 40] was to design a control strategy robust to uncertain transport delay.

Li et al. in [42] implement methods based on linear control theory. A Kalman filter is used to provide optimal state prediction for an end-to-end connection. However these methods are limited and their effectiveness is questioned as they rely on linear models and linear control theory.

It is commonly understood that computer networks are highly non-linear systems, especially during periods of high congestion, but non-linear network models are intricate and difficult to develop. Consequently system identification techniques are a likely alternative.

Model predictive control (MPC) techniques based on identified system models are presented in this thesis. Basic principles of the techniques used in this work are described in [43, 44, 45]. MPC schemes are used in many engineering fields. In [46, 47, 48] receding horizon techniques are applied in the aerospace field. Multi-step prediction equations are derived based on linear system identification models. The desired control effort is calculated as the solution of the equation which minimizes the difference between predicted and desired output. Similar techniques are developed in this research for distributed real-time multimedia applications.

C. Proposed Solution

Several goals must be accomplished to address the stated research objectives. For the present work the end-to-end system is modelled using a fluid model. Packet streams are modelled as fluid flows, in order to cope with the complexity of the problem. Simple conservation equations are analytically derived to build up the end-to-end model. Cross-flow traces are extracted from NS-2 simulations [49], and combined with the derived model to compose a realistic network model as the model response is as realistic as the cross-flow input to it. SIMULINK and MATLAB are used to simulate

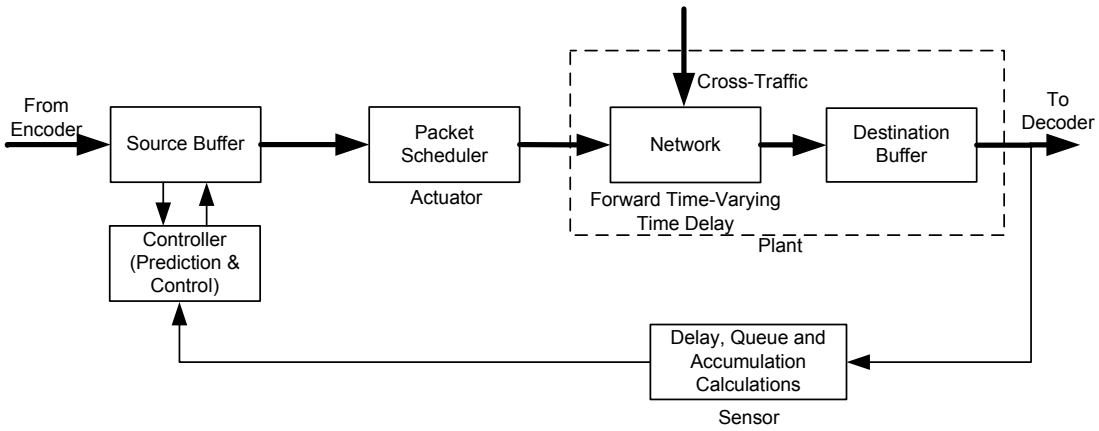

Fig. 1. A Block Diagram of the Proposed System.

the end-to-end fluid model. Control efforts are calculated based on various end-to-end network parameters as feedback signals. The controlled models are investigated and compared to the initial uncontrolled model. The effects of the feedback signal on the end-to-end characteristics, such as buffer level, packet losses and media playback are explored. Different control strategies are developed and compared.

Figure 1 shows the proposed system configuration in a system theoretic framework. Destination buffer level, end-to-end delay and packet accumulation signals are used in adaptive, predictive or reactive control algorithms with the intent to control the system plant, which is the combination of the best-effort network and the destination buffer. The source buffer is used for storing data, allowing time-variable control actions. The sensor is a mechanism which observes the buffer occupancy, end-to-end delay, flow accumulation and other end-to-end parameters if needed, and sends feedback signals to the controller. Then the controller performs the specific control function and imposes upon the actuator to adapt the actual sending flow-rate by scheduling the packet departures appropriately. The function of the actuator is performed by a packet scheduler.

D. Contributions of This Work

This thesis proposes an application layer control scheme, and demonstrates that QoS can be improved for real-time multimedia applications with hard constraints over current practice of sending packets at rates equal to the encoder bit rate.

This thesis develops a systematic media application level modeling and control design framework using fluid-based models. It establishes the mathematical background for modeling and control of much more complicated network scenarios.

It develops adaptive predictive techniques, which can be the cornerstone for more advanced and effective control methods, which would make internet an effective means for real-time communication.

Specifically the following contributions are made:

1. A fluid-based model of the end-to-end system for a single multimedia flow is analytically derived and simulated. Although a simple network scenario is assumed, where packet flow travels over a unique path, the derived models can be the elements of much more complex networks. This work can be advanced into a "mini" network simulator.

2. Fluid-based reactive and predictive flow control methods are developed at the application level over unreliable transport protocol in best-effort networks and shown to improve the QoS for real-time applications.

3. System identification techniques are used to create input-output relationships between several best-effort network measurements. Model Predictive Control methods are developed based on these empirical models. The techniques of system identification as well as the Model Predictive Control methods, can be beneficial when implemented over much more complicated networks, where

data-driven modeling is necessary.

This research demonstrates that source-based adaptive information flow control implemented over best-effort networks, can improve the QoS of real-time media applications as perceived by the user. In particular, by adding a source buffer the controller shifts some of the overall end-to-end delay present at the destination buffer to the source buffer allowing control of packets destined to be dropped by the network. The overall impact of adding a source buffer, is to increase somewhat the total packet end-to-end delay, in order to allow reduction in loss rates over best-effort networks.

E. Thesis Overview

In Chapter II, the major assumptions and constraints of the research problem are discussed. Chapter III outlines several methods of modeling the system from an end-to-end systems engineering perspective. Control algorithms are developed in Chapter IV and simulation results presented in Chapter V. The work is summarized, conclusions are provided, and recommendations for future work in this area are included in Chapter VI.

CHAPTER II

COMPLEXITIES OF END-TO-END FLOW CONTROL IN DISTRIBUTED REAL-TIME APPLICATIONS

In this chapter, several issues regarding the system of interest are addressed. The end-to-end system consists of several separate subsystems with different dynamic characteristics, each one contributing to the complexity of the total system. The end-to-end system is described and the difficulties of applying feedback control, are elaborated. Several assumptions are also made.

A. Issues Regarding the End-to-End System

Defining the distinguishing characteristics of the system under consideration is appropriate before developing relevant dynamic models and designing controllers. The proposed controller configuration is also described at some high level.

1. Fluid Level vs Packet Level Simulation

Traditionally, packet-level simulations have been widely used for performance evaluation of packet-switched computer networks. However, due to the fast growth of data communication and the increased complexity of the networks, packet-level approaches are computationally expensive and most likely impossible. This is the reason that more efficient simulation techniques have been sought. As mentioned in [9], methodologies that speed-up computer network simulations can be classified into three major categories as follows:

1. Computational power: It includes methods which propose the use of faster and more powerful computing machines.

2. Simulation technology: Methodologies utilizing new advanced algorithms are included.

3. Simulation model: This approach is to use models with higher level of abstraction. Making simplified assumptions about the real system, simplifies the simulation process and improves its efficiency. The fluid model is one of the techniques used in the literature.

This research is based on fluid models of packet flow. The data streams are assumed to be continuous-time flows. Network traffic is modeled in terms of continuous fluid flow rather the discrete packets. In this way, large number of packets are represented with a single fluid chunk, significantly decreasing the processing needed for simulating the model. The important tradeoff is the simulation accuracy obtained by the more abstract fluid model. However, it has been found that fluid-based simulators generally outperform packet-level simulators and the error introduced due to the high level of abstraction, is relatively small [9].

2. Unreliable Transport Protocol, Ordered Flows and Packet Losses

Interactive media applications are investigated in this research. A good example of this type of applications is Voice over IP (VoIP) and video conferencing. VoIP can be defined as the ability to deliver voice packets over IP-based data networks in real-time. Voice and video services have strict QoS requirements on delays and losses as compared to other applications. This is the reason that these applications use UDP instead of TCP as the transport protocol to transfer data. Although TCP guarantees complete reliability of information to the end-user because of the retransmission of the lost packets, the QoS specifications of the application deteriorate when TCP adjusts itself to network congestion. Another reason UDP is preferred over TCP is that the

header size for UDP is much smaller than that for TCP. In this way for the same payload, UDP sends smaller packets into the network which results in more efficient bandwidth utilization. In addition to the above, overhead for TCP is more than UDP since TCP uses extra processing time for setting up connections or for error checking.

On the other hand, in attempting to study and model the open-loop end-to-end flow dynamics, one must consider flows transported using the UDP as the transport protocol. TCP uses a feedback signal from the destination transport layer to the source transport layer and therefore the TCP end-to-end dynamics are not considered open-loop network dynamics, containing a transport layer feedback loop, with certain inherent performance limitations.

For the current development the flow models are based on the assumption that the applications are interactive. Also the open-loop uncontrolled flow dynamics are investigated and compared to the closed-loop scenarios. Therefore, it should be a natural choice to assume UDP as the transport protocol used at the source to be controlled.

Real-time applications are sequence sensitive. In some other applications, such as e-mail, sequence does not matter since the content is not intended for real-time use. Packet sequencing errors or flow reversal effects exist when the delay experienced by a specific flow, changes faster than the time is advanced. In such case, segments of the specific flow arrive in different order than the order they were sent. This case occurs generally when multiple simultaneous flow paths are available between a source and a destination. If flow reversal is considered, the control problem becomes much more complicated, if not infeasible. In this research, a single path between the source and the destination is assumed. Consequently there is no possibility of flow reversal.

Another common occurrence in packet-switched networks is packet loss. Figure 2, shows a system when losses occur. Dampers and resistors describe losses in mechanical

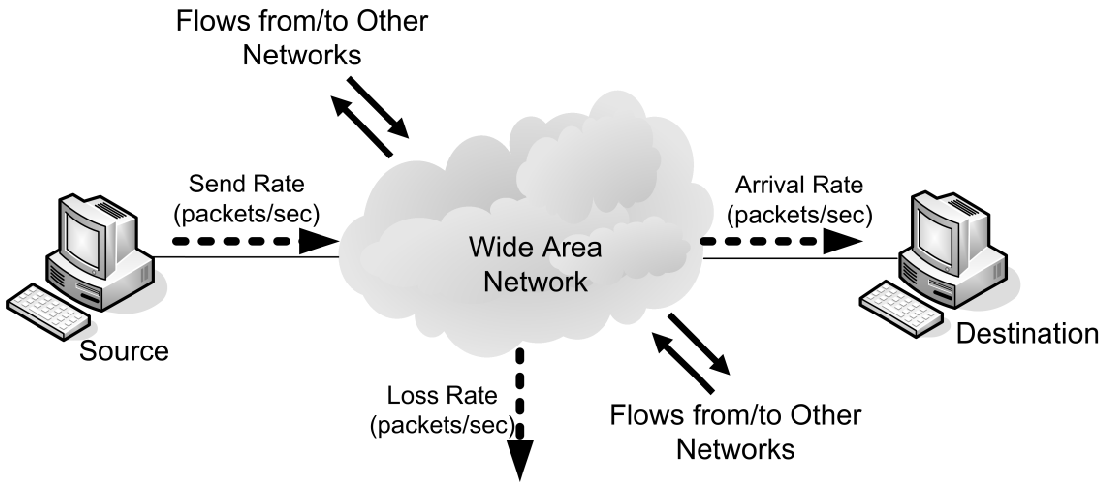

Fig. 2. End-to-End System Showing the Send Rate, Arrival Rate and Packet Loss Rate of a Single Flow.

and electrical systems. Packet losses along with packet delays, are the two major measures of performance in best-effort networks. Losses occur when the router queues are full. The router queues are simply buffers with finite capacity. In periods of high congestion when the incoming rates are larger than the router service capacity, then the queues overflow. In that case, there is simply no space for new flows and parts of them are dropped. In this thesis losses are also modelled in the assumed flow as will be shown in the next chapter.

3. The Time-Varying Nature of the End-to-End Time-Delay

The end-to-end delay is defined as the difference between the time when packet exits the application layer at the source and the time when packet enters the application layer at the destination. This is the one-way end-to-end delay. The end-to-end delay is the major cause of all the problems packet transport experiences. If the delay was constant, then the problem of improving the QoS would be trivial. Several factors

affect the delay. The major causes of delay in the Internet are mainly four: Nodal processing, transmission, queuing and propagation, each having specific causes and effects. The processing delay is defined as the time associated with checking of the bit-level errors in the packet and examining the packet headers. The transmission delay is defined as the time taken to transmit all the bits of a packet into the link. It can be obtained by dividing the length of the packet by the transmission capacity of the link. The propagation delay is defined as the time required by a bit to propagate from the beginning of the link to the end of the link. This type of delay depends on the physical medium of the link and the distance between the two connected nodes. The last component is the queuing delay and it is defined as the residence time of the flow in the queue. It's simply the time flow waits in the queue to be transmitted onto the next link. This delay, clearly depends on the packets that are already in the queue waiting to be transmitted as well as on the queue size, among others. In this research it is assumed that the first two components, i.e. processing and transmission are constant and negligible compared to the last two. In fact it can be assumed that all three components are lumped into one single component whereas queuing delays are treated separately. Therefore the end-to-end delay can be expressed as

$$\tau(t) = \tau_c + \tau_q(t), \tag{2.1}$$

where τ_c, is the constant propagation delay and $\tau_q(t)$, the time-varying queuing delay.

4. The Network Topology

In order to define the problem as well as demonstrate the solution to the problem, a network topology must first be selected. Consider a data stream that is transported via a packet-based medium, such as a Wide Area Network (WAN). In a realistic

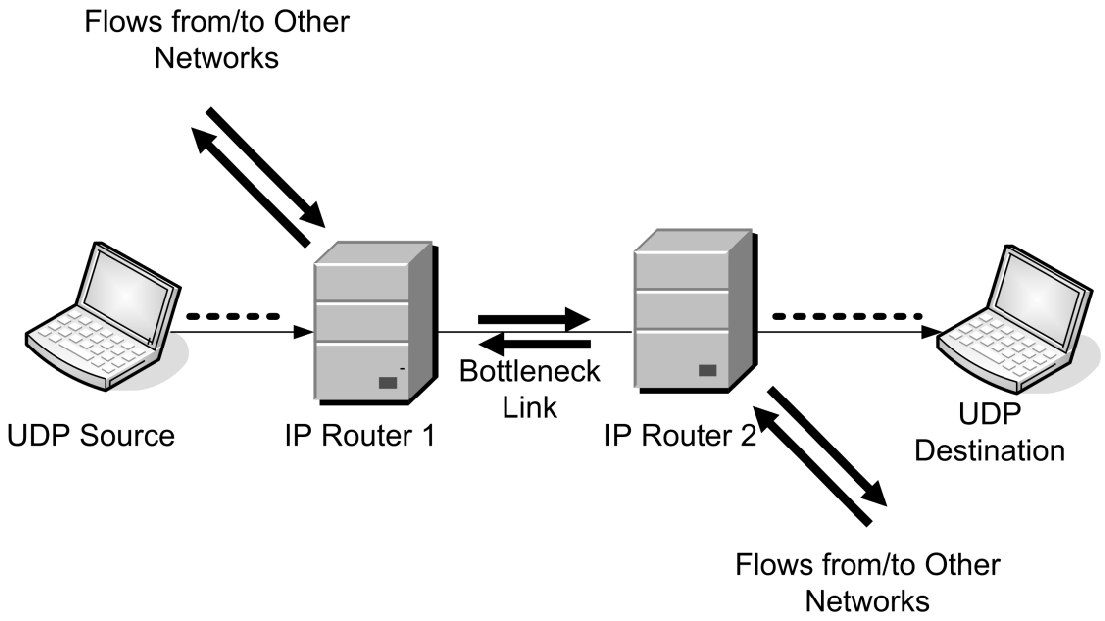

Fig. 3. Simplified Network Topology.

network scenario the WAN would include thousands of nodes and links. However, the simplified WAN is modelled as a single congested queue, as shown in Figure 3.

The cross-traffic is modelled using the Network Simulator (ns-2) and it represents the sum of all the sending rates of all other sources sending traffic through the network router connecting the bottleneck link. It can be seen that the network is simplified significantly by using a single bottleneck link. The maximum total flow rate that can traverse the link is equal to its capacity and in this study is equal to the link bandwidth. The Router 1 is the crucial node of the system. All flows meet at this router, which is responsible for forwarding the flows to the bottleneck link, according to the selected scheduling scheme of the router. In this work, it is assumed that the multiplexer's scheduling scheme is first-come, first-served (FIFO). Consider a case in which the sum of the queue input rates exceeds the capacity of the bottleneck

path. Then, the queue will start building up and eventually packet drops are likely to occur. The second router's objective is simply to route each flow to its final destination. Neither flow losses nor queuing delays are associated with the second router. The topology will be discussed in more detail in Chapter V.

B. The End-to-End Control Problem

There are multiple complications that should be clarified, before the control design. There are three objectives to be accomplished as mentioned in the first chapter. However, each one of the objectives is most of the times, competing against the other two.

1. Complexity of the Control Problem

Reconsider the objectives of the control problem, which are:

1. Minimize the playback start time.
2. Reduce flow losses compared to the case with no control.
3. Prevent disruptions at destination buffer during media playback.

Due to the real-time nature of the application, in order to implement any kind of control, a source buffer is must be added at the application layer as will be shown in Chapter III.

Figure 4, depicts the system when no application feedback is used in the presence of UDP and the same system when an application layer controller is implemented. As one can see, the addition of the source buffer at the application causes an additional, inevitable end-to-end delay due to the initial buffering at the source. There will always be a small constant delay, necessary for the source buffer to reach a desired level. This

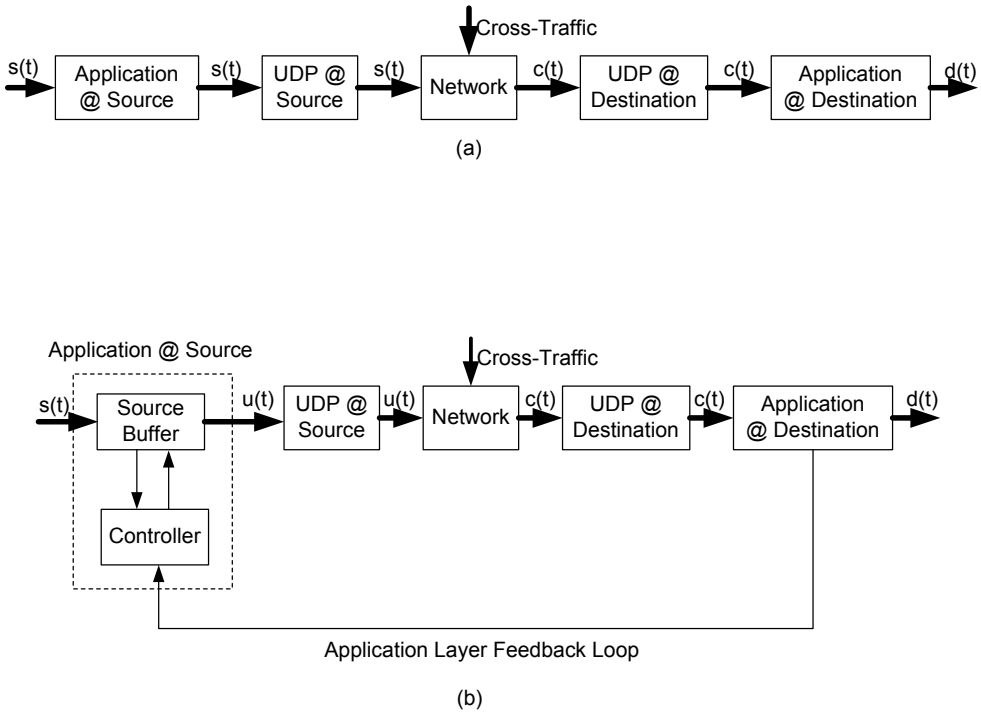

Fig. 4. (a) End-to-End System with No Control and (b) System with Application Layer Feedback.

will be further discussed when the source buffer is modelled. Therefore, the goal of the controller will be to achieve the other two objectives while keeping the dead-time as close as possible to the dead-time of the uncontrolled case. However, there is a case where minimizing losses may reduce dead-time. This is when a controller achieves loss reduction during the dead-time. Then as the throughput achieved will be higher, the arrival rate will be higher and therefore the destination buffer can reach the initial desired level faster as compared to the uncontrolled case where losses are more and therefore throughput is less. This can be achieved when the losses prevented during dead-time are considerable.

The real complexity of the control problem comes into picture when flow loss reduction and playback disruption prevention become the controller objectives. Consider again the open-loop, uncontrolled case. It is assumed that the application sends packets at a constant rate which in fluid model is represented by a pulse function. The destination expects to receive the same constant rate at the destination somewhat delayed. However, the network dynamics (the time-varying time-delays and the losses), alter the flow and the destination client does not receive the same constant flow rate simply shifted in time.

The dynamics of the network are directly reflected by the arrival rate, which in this study is assumed to be the rate at which the packets exit the network and enter the destination buffer. Queuing delays compound the complication with an elastic effect in time. As the end-to-end delay increases, the flow arrival rate is less than the corresponding flow send rate. Similarly, as the end-to-end delay decreases the flow arrival rate increases relative to the flow send rate. Since the arrival rate is the rate at which the flow enters the destination buffer, the buffer is likely to empty when the arrival rate decreases. This means that when the queuing delays are high the destination buffer at empties and the playback will is disrupted. In addition to the

effects leading to disruptions due to empty destination buffer, flow losses are caused by large delays. When the router buffer is saturated, the queuing delay reaches its maximum value as the queuing delay is directly proportional to the router queue level. Flow losses occur when the buffer is completely filled up. The relation between delay and flow loss is qualitatively simple, but mathematically complex.

Consider the second control objective. That is, a controller should send the packets in such a way that flow losses are reduced. Intuitively, the control effort should be such that higher controller rates are present when the end-to-end delay is low and the opposite when the delay is measured or anticipated to be high. Qualitatively speaking, such a control strategy should be able to reduce flow losses compared to the uncontrolled case.

However, the third objective can not be guaranteed. In order to prevent playback disruptions the destination buffer should be kept at a constant, desired level so that it will always contains excess packets. The destination buffer empties when the time-varying queuing delay is high. Therefore, if a controller is designed to prevent playback disruptions by keeping the destination buffer level constant, then the send rate would be higher when the end-to-end delay is high and lower when the delay is low. As a result, the last two objectives are in conflict. If the objective of the controller is to only keep the buffer level constant, the tradeoff would be an increase in flow losses. Conversely, if the controller objective is to only reduce flow losses then playback disruptions are more likely to happen.

Consequently, a compromise solution should be sought for. A control function that has weighted terms which compensate for both losses and destination buffer level should be attempted. Feedback signals which can be utilized for flow loss prevention are the end-to-end delays and flow accumulation, whereas for playback disruption prevention, the destination buffer level should be used. This will be further explained

when the control strategies are defined. The notion of packet or flow accumulation will also be explained. For now, consider accumulation to be a signal proportional to the end-to-end delay.

In addition to the above arguments, the source buffer places an additional constraint on the control effort. The controller will calculate the ideal send rate to be sent over the network. However, if the source buffer is empty the control effort which is the output rate of the source buffer can not exceed the input rate. Therefore, even if a large control effort is called for, there may not be enough information contained in the source buffer to satisfy the controller. This means that the ideal control effort, $u_{id}(t)$, is not always equal to the actual send rate $u(t)$. In the case of non-real-time application content transport [29], the controller has all the content available at its disposal and henceforth the source buffer constraint, is no longer present, rather it is replaced by the available bandwidth at the source.

2. Proposed System Configuration

A block diagram depicting the uncontrolled dynamics of the end-to-end system is shown in Figure 4. A feedback loop can be applied at the application layer to enable or enhance the application level QoS of the flow. Note, that the network is part of the system dynamics and it is modeled as single congested queue, as mentioned earlier.

Control is implemented by applying various feedback and predictive control concepts to the system dynamics. Figure 5 shows a block diagram of the proposed system architecture.

It can be seen that multiple feedback loops are used to achieve the controller objectives. The controller shown in Figure 5 handles prediction, buffer level regulation and flow loss prevention tasks.

The destination buffer holds the media content for playback. It is modeled as

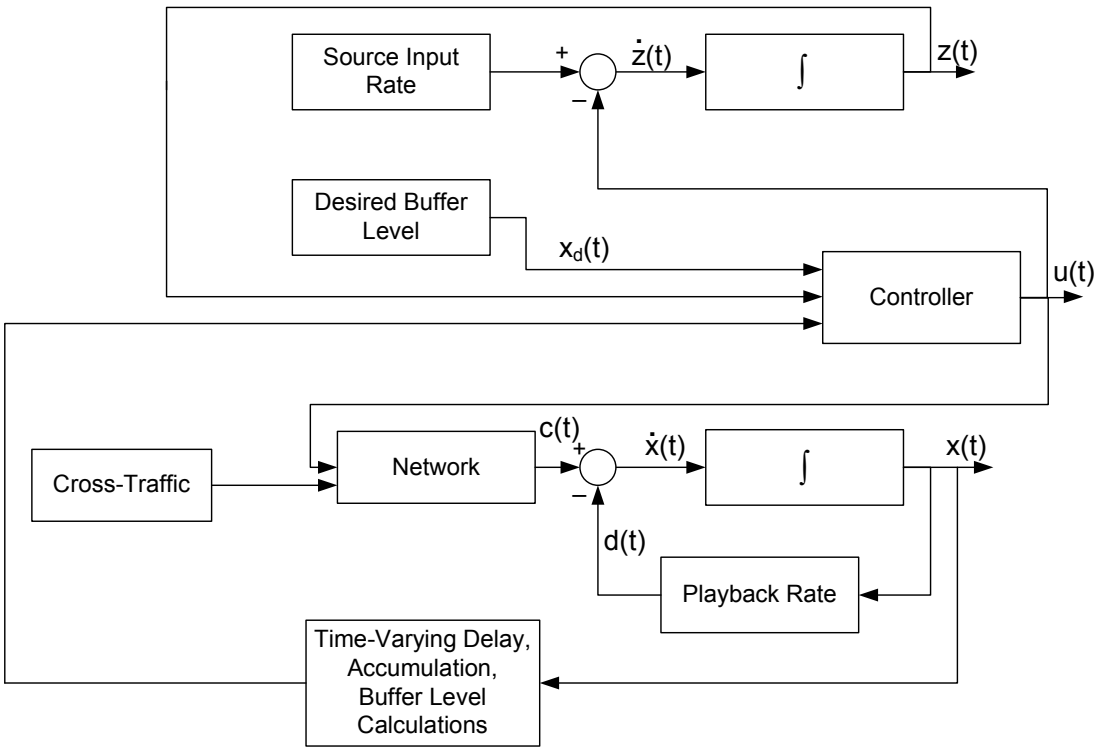

Fig. 5. A Block Diagram of the Proposed Control System Using Feedback of Delay, Accumulation and Buffer Level.

an integrator and it is part of the system dynamics. The destination application also records the flow arrival times, the buffer level, flow accumulation, and delay information for feedback to the source. Similarly, the source buffer is described as an integrator and its level is fed forward to inform the controller about content that are stored in it and is available for forwarding into the network. In real-time applications, the source buffer level defines the maximum send rate that can be transmitted over the network.

The network is modeled as a single queue. It is mathematically defined as an integrator with high and low level limits, since the queue level can not be negative or exceed a specific high limit, which is the queue capacity. The cross traffic is modelled using ns-2. Cross-traffic trace files from ns-2 are imported into the fluid model. The ns-2 simulation scenarios reflect the same condition studied by the fluid model.

The end-to-end application control approach described in this chapter is advantageous because it could be possible to implement it for application QoS without requiring any changes to the network infrastructure. No information from the network architecture is needed; all required feedback information can be obtained from the end-points.

C. Assumptions

The following major assumptions are made throughout this research:

- Packets are assumed to constitute a continuous flow, rather than discrete events.
- Only one and the same congested queue is assumed present on the end-to-end path. As a result of this assumption flow reversal (or packet sequencing error) can not occur.
- The network queues are assumed to be First-In First-Out (FIFO).

- The real-time application considered is assumed to generate content at a constant rather than variable bit rate.
- The controlled flow to cross-flow ratio is approximately 1 to 10. Typically the ratio is on the order of 0.1% or less.
- The cross-flow is such, that the end-to-end delay varies slowly.
- Feedback signals from the destination back to the source are not assumed to be delayed, rather they are assumed to be available instantaneously. Information from the destination may take as long or longer to arrive at the source than a forward signal took to arrive at the destination. Therefore, destination information of present time is available for use at the source application layer for control purposes.
- Some of the control strategies as will be shown in the next chapters, are structured based on the end-to-end delay measurements and predictions. It is assumed that precise packet delay measurements and predictions are available at both source and destination. Problems related to clock synchronization are ignored.

D. Chapter Summary

In this chapter, some classifications of end-to-end system are made and issues and constraints related to computer networks are introduced. Application level flow rate control is discussed with an intent to develop some possible control structures. The major assumptions made in this research are listed and some initial groundwork is laid for system modeling and control synthesis.

CHAPTER III

DYNAMIC MODELING OF THE END-TO-END FLOW TRANSPORT SYSTEM

Before developing the control schemes, it is appropriate to model the system of interest. It is extremely difficult to conduct systematic control design using packet-level dynamic models of applications because such models can not be expressed in the form of differential equations. As a result, in this thesis, fluid models of packet flows are developed. These fluid packet flow models are treated in the continuous-time domain. The end-to-end system consists of several subsystems, and the dynamic models of each one are developed.

A. Source Buffer Model

Because the desired objective of the controller demands that the control effort be time-varying while the application is assumed to generate content at a constant rate, a source buffer is necessary. Therefore, a buffer is placed between the signal source and the controller. The limited buffer size places constraints on the control effort as the source buffer capacity places lower and upper limits. However, in this development the buffer upper constraint is not considered, as the major problem appears to be the lower limit of an empty buffer. The dynamic model of the source buffer is a simple conservation relation. A block diagram of a system with a source buffer is shown in Figure 6.

The source buffer level can be described by the state variable $z(t)$. The buffer removal rate is defined as $u(t)$. The control effort calculated by the controller is denoted as $u_{id}(t)$. In the ideal case $u(t)$ is equal to $u_{id}(t)$. However, this is not always the case as $u_{id}(t)$ is also a function of the source buffer level which places constraints on the control effort. For example, if the controller calls for $u_{id}(t)$, then the send rate

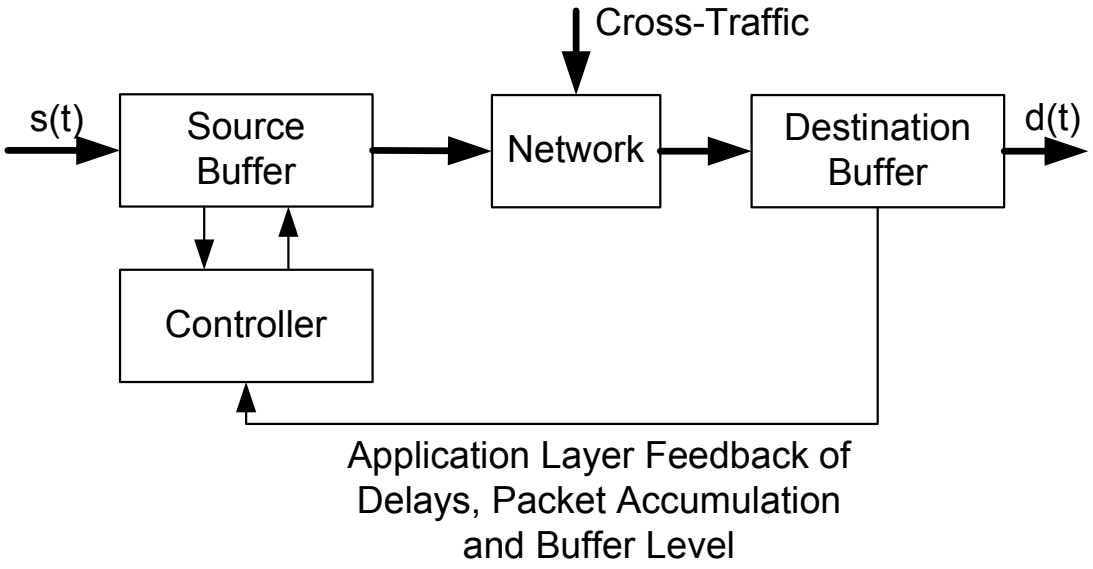

Fig. 6. Block Diagram of the System Showing a Source Buffer.

will be equal to $u_{id}(t)$ for a certain time period, only if the source buffer has enough reserved content for the duration of this period. Otherwise the send rate $u(t)$ will be less than $u_{id}(t)$ and limited by the buffer input rate. The buffer input rate $s(t)$, is the rate at which information is made available to the system by the application. A conservation relationship can be formulated where $z(t)$ is described by a simple integrator as follows:

$$\dot{z}(t) = s(t) - u(t). \tag{3.1}$$

The source input rate as well as the destination buffer removal rate are assumed constant and equal. Any increase in the destination buffer will demand a decrease in the source. Therefore a large control effort will be called for at the source to compensate for the empty destination buffer. The source buffer must have reserved information in it to be able to provide for the called control effort. For that reason

a minimum constant delay is required for the source buffer to reach an initial level to be able to accommodate for any increases in the control effort. If this level is not high enough, the source buffer will empty and the control will be limited to the source constant input rate.

B. Network Dynamic Model

Modeling a network is not an easy task. A simple network scenario is assumed for this research. The source to be controlled as well as all other sources are connected to the congested router. The router is a single, FIFO queue where all flows are assumed to be treated equally. The simulation network topology is shown in Figure 3.

1. Time-Varying Time Delay Model

Time-varying time delay is detrimental for systems generally and especially for packet-switched computer networks. Delay jitter which is the variation of the delay magnitude is the main cause of packet accumulation in the queues and consequently of packets arriving late or never arriving at the destination.

However, it is important to develop and investigate a model for the time-varying time-delay. Otherwise, the network model can not be considered as realistic and the modelled dynamics will not accurately reflect the dynamics of a real network. It is appropriate to define the end-to-end delay function in order to be able to relate the network output flows to past values of the input flows.

We already defined the input flow as $u(t)$ which is the source buffer removal rate. All other sources are connected to the network router as well. The send rate of the source i is defined as $u_i(t)$. It is assumed that n other sources are connected to the queue and each one generates one flow. Then, the total cross-traffic flow rate entering

the router is

$$u_{cr}(t) = \sum_{i=1}^{n} u_i(t), \tag{3.2}$$

which allows us to treat the sum of all other flows as a single flow. The rate at which the signal under investigation arrives at the destination is defined as $c(t)$ and is the rate at which the flow exits the network. Similarly we can define the total cross-traffic flow arrival rate as

$$c_{cr}(t) = \sum_{i=1}^{n} c_i(t). \tag{3.3}$$

Then the total rate entering the queue is $u_{cr}(t)+u(t)$ and the total rate departing from the queue is $c_{cr}(t) + c(t)$. Therefore, the the router queue dynamics can be expressed as

$$\frac{dq(t)}{dt} = u_{cr}(t) + u(t) - c_{cr}(t) - c(t), \tag{3.4}$$

where $q(t)$ is the instantaneous router queue (buffer) level. The router buffer capacity is assumed to be q_{max}. The queue can not be negative and also can not exceed the capacity of the buffer. That is mathematically expressed as $0 \leq q(t) \leq q_{max}$. The bottleneck link capacity is defined as C. The total outflow rate can not be negative and it can not be more than the link capacity rate. That is $0 \leq c_{cr}(t) + c(t) \leq C$.

The queuing delay or the average residence time in the queue experienced by the flow arriving at time t, [12], can be defined as

$$\tau_q(t) = \frac{q\left(t - \tau_q(t)\right)}{C}. \tag{3.5}$$

Equation (3.5) can be solved iteratively. To explore the behavior of the rate of change of the delay one should differentiate Equation (3.5) with respect to the arrival time. Let's define as $\dot{q}\left(t - \tau_q(t)\right) = \frac{dq(t-\tau_q(t))}{d(t-\tau_q(t))}$ and $\dot{\tau}_q(t) = \frac{d\tau_q(t)}{dt}$. Then it is obtained

$$\frac{d\tau_q(t)}{dt} = \frac{\dot{q}\left(t - \tau_q(t)\right)\left(1 - \frac{d\tau_q(t)}{dt}\right)}{C}. \tag{3.6}$$

Let

$$\gamma(t) = \frac{\dot{q}(t - \tau_q(t))}{C}. \tag{3.7}$$

Substituting (3.7) into (3.6), the latter is converted into

$$\dot{\tau_q}(t) = \frac{\gamma(t)}{1 + \gamma(t)}. \tag{3.8}$$

To prove that definition (3.5) is valid for the case of a single end-to-end path one should express the flow send time, θ, in terms of arrival time t and delay $\tau_q(t)$, as in [29]. Then

$$t = \theta + \tau_q(t). \tag{3.9}$$

Taking the derivative of (3.9) it can be obtained that,

$$1 = \dot{\theta} + \dot{\tau_q}(t). \tag{3.10}$$

If the derivative of the send time with respect to the arrival time is negative, then the arrival time is decreasing. But in the current development it is assumed that flow order reversal is not possible to occur. Therefore, for $\dot{\theta}$ to be positive,

$$\dot{\tau_q}(t) < 1. \tag{3.11}$$

Equations (3.8) and (3.11) imply that

$$\gamma(t) > -1, \tag{3.12}$$

which in turn, implies that

$$\dot{q}(t - \tau_q(t)) > -C. \tag{3.13}$$

This always the case in view of Equation (3.4). As one can see, the largest drop rate in the queue level occurs when the total send rate $u_{cr}(t) + u(t)$ is zero. In this case, the queue level drop rate will be equal to the link capacity, and then

$$\dot{q}(t - \tau_q(t)) = -C. \tag{3.14}$$

Therefore, to assure that inequality (3.13) is always valid, the total input rate should never drop to zero. Then Equation (3.4) is always valid and can be used for modeling the queuing delay when single path is assumed. In the case when total inflow becomes zero, then $\dot{\theta}$ becomes zero, which implies from (3.10) that

$$\dot{\tau_q}(t) = 1. \tag{3.15}$$

From (3.7) and (3.14) it is obtained

$$\gamma(t) = -1, \tag{3.16}$$

which implies from (3.8), that $\dot{\tau_q}(t)$ is infinite.

This indicates an inconsistency between the two values of $\dot{\tau_q}(t)$. Therefore, it can be concluded that (3.5) implies that $\dot{\tau_q}(t) < 1$.

2. Congested Network Router - Output Flow Splitting - Loss Model

A decisive part of the model is definitely the network. Equation (3.4) is not valid for all cases. The behavior of the rate of change of the queue level when the queue is close to its upper and lower limits should be explored. The model should be able to

capture the effects of the losses, which enter the problem when the router queue level reaches its capacity and therefore $q(t) = q_{max}$. Then the router buffer is saturated and no space for new packets exists. Packet drop is the immediate consequence.

In order to be able to address the end-to-end problem the output flow of the controlled source should be modelled. Since all other sources are also connected to the queue the flows have to traverse the common bottleneck link. The service capacity should be somehow divided among the competing flows.

In order to describe $c(t)$ and $c_{cr}(t)$ it is necessary to consider different conditions. Define $\dot{l}(t) = \frac{dl(t)}{dt}$, as the loss rate of the flow considered due to finite buffer space and $\dot{l}_{cr}(t)$ the loss rate of the cross traffic flow. The following network conditions are considered, depending on the queue level:

1. The first case is when $q(t) = 0$ and $u(t) + u_{cr}(t) < C$. Then there is no accumulation in the router queue and the total input rate is less than the service capacity C. Then the following equations apply

$$\begin{aligned} \tfrac{dq(t)}{dt} = u_{cr}(t) + u(t) - c_{cr}(t) - c(t) = 0, \\ c(t) = u(t), \\ c_{cr}(t) = u_{cr}(t), \\ \tfrac{dl(t)}{dt} = 0, \\ \tfrac{dl_{cr}(t)}{dt} = 0. \end{aligned} \tag{3.17}$$

   Equations (3.17) simply say that in this case outflow is equal to inflow at time t. No accumulation occurs in the queue since the total inflow is less than the link capacity.

2. The second case is when $0 < q(t) < q_{max}$. In this case flow accumulation occurs in the router buffer and the link service capacity is divided among the competing

flows in proportion to their sending rates. Let the proportion of the sending rates be defined as

$$p(t) = \frac{u(t)}{u_{cr}(t) + u(t)}. \tag{3.18}$$

Under these conditions,

$$\begin{aligned} \tfrac{dq(t)}{dt} &= u_{cr}(t) + u(t) - c_{cr}(t) - c(t), \\ c(t) &= p\left(t - \tau_q(t)\right) C, \\ c_{cr}(t) &= \left(1 - p\left(t - \tau_q(t)\right)\right) C, \\ \tfrac{dl(t)}{dt} &= 0, \\ \tfrac{dl_{cr}(t)}{dt} &= 0. \end{aligned} \tag{3.19}$$

The output flow of the queue at time t is a function of the input flow rates that enter the queue at time $t-\tau_q(t)$. No flow is lost since the queue is not saturated.

3. The last case is when $q(t) = q_{max}$. In this case the queue level reaches its upper limit and the router buffer overflows. There is no space for the incoming flows and part of the flow is lost. This case is further separated into two other subcases, as follows:

   (a) If $q(t) = q_{max}$ and $u(t)+u_{cr}(t) > C$, then the total input rate is greater than the service capacity while the queue is saturated. This can be expressed as

$$\begin{aligned} \tfrac{dq(t)}{dt} &= 0, \\ c(t) &= p\left(t - \tau_q(t)\right) C, \\ c_{cr}(t) &= \left(1 - p\left(t - \tau_q(t)\right)\right) C, \\ \tfrac{dl(t)}{dt} &= p(t)\left(u_{cr}(t) + u(t) - C\right), \\ \tfrac{dl_{cr}(t)}{dt} &= \left(1 - p(t)\right)\left(u_{cr}(t) + u(t) - C\right). \end{aligned} \tag{3.20}$$

(b) If $q(t) = q_{max}$ but $u(t) + u_{cr}(t) < C$ then no flow is lost, and the case is similar to case 2. That is,

$$\begin{aligned} \tfrac{dq(t)}{dt} &= u_{cr}(t) + u(t) - c_{cr}(t) - c(t), \\ c(t) &= p\left(t - \tau_q(t)\right) C, \\ c_{cr}(t) &= \left(1 - p\left(t - \tau_q(t)\right)\right) C, \\ \tfrac{dl(t)}{dt} &= 0, \\ \tfrac{dl_{cr}(t)}{dt} &= 0. \end{aligned} \tag{3.21}$$

The last case can be visualized more easily when it is compared to a simple hydraulic system. Consider a water tank which has a drain valve that allows water to flow out of the tank. A source lets water flow into the tank. If the input rate is larger than the output rate, the tank will fill up. If the input is still more than the output, then the difference between the two rates will spill out of the tank. If the input rate becomes less than the output, then no water will be lost and the tank level will drop.

The model for the flow of interest as well as the router queue dynamics can be summarized as follows

$$\frac{dq(t)}{dt} = \begin{cases} 0, & \text{if } q(t) = 0 \text{ and } u(t) + u_{cr}(t) < C, \\ 0, & \text{if } q(t) = q_{max} \text{ and } u(t) + u_{cr}(t) > C, \\ u(t) + u_{cr}(t) - C, & otherwise. \end{cases} \tag{3.22}$$

$$\frac{dl(t)}{dt} = \begin{cases} p(t)\left(u(t) + u_{cr}(t) - C\right), & \text{if } q(t) = q_{max} \text{ and } u(t) + u_{cr}(t) > C, \\ 0, & otherwise. \end{cases} \tag{3.23}$$

Finally, the outflow rate from the queue, which is the arrival rate at the destina-

tion buffer is given by

$$c(t) = \begin{cases} u(t), & \text{if } q(t) = 0, \\ p\left(t - \tau_q(t)\right) C, & otherwise. \end{cases} \tag{3.24}$$

The above equations constitute the complete model of the flow under consideration. The flow conservation imposes that the integral of the send rate must be equal to the integral of the loss rate plus the integral of the arrival rate over a finite time period.

C. Destination Buffer Model

The destination consists of the destination buffer, where the flow which arrives at the rate $c(t)$ is stored and is played back. The playback is the buffer content removal rate. Smooth playback, with no disruptions is one of the control objectives. Ideally the playback should be equal to a delayed version of the source buffer input rate function.

The destination buffer level is defined as, $x(t)$. The removal rate from the buffer, $d(t)$, is also time dependent. The rate at which signals arrive at the destination, $c(t)$, is a function of the delay $\tau_q(t)$ and the send rate $u(t)$ as shown in previous sections. A conservation relationship can be formulated for the destination buffer where $x(t)$ is described as the state of a simple integrator similarly as in the source buffer,

$$\dot{x}(t) = c(t) - d(t). \tag{3.25}$$

The instantaneous destination buffer level for any time t is simply the arrival rate of the flow that came into the buffer at time t, less what is removed due to the playback removal rate at the same time instant. The buffer level $x(t)$ is a signal that will be used as a feedback for the controller and it is important that it is kept within

some limits in order to prevent disruptions.

The buffer content removal rate model is dependent on the destination buffer level. This relationship is nonlinear and it can be described by the function

$$d(t) = f(x(t)). \tag{3.26}$$

The destination buffer must be kept within certain constraints to prevent overflowing or emptying. In addition to these constraints, a softer constraint is also set; that of a desired setup buffer level. Content removal from the buffer should not start until the buffer initially reaches a specific level. The more this initial level is increased the less likely is for disruptions to occur but system dead-time also increases by the time needed for the buffer to reach this initial level.

After that, playback can continue as long as the buffer remains within the upper and lower bounds. If the upper and lower constraints are met, then playback will not be interrupted and the end-user will perceive good QoS.

Modeling the content playback rate is done by applying the already specified limits in mathematical form. If the destination buffer level is less than or equal to the lower bound or greater than or equal to the upper bound, then playback will stop. Again, assume that the desired constant playback rate at the destination is $d_{desired}$, equal to the source buffer input rate. Let the upper and lower buffer constraints be x_{max} and x_{min}. Also, let the level required for transition from the buffer setup phase to the normal playback phase be x_{on}. This means that the buffer level may be above the lower bound x_{min} but playback is not commenced because the desired setup buffer level x_{on} is not yet achieved. Consider S to be a Boolean operator or a switch that determines if the setup phase is complete. That is, it is true if the x_{on}

has been reached. Then the following model applies:

$$d(t) = \begin{cases} d_{desired}, & \text{if } x(t) > x_{min} \text{ and } x(t) < x_{max} \text{ and } S = true, \\ 0, & otherwise. \end{cases} \tag{3.27}$$

The nonlinear relation for the destination buffer removal rate is described by (3.27).

The total system model is a combination of each of the models described previously. Equations (3.1), (3.5), (3.22), (3.23), (3.24), (3.25) and (3.27), constitute the model used to design and test the flow controllers.

D. Software Used for Simulation Studies

The continuous-time fluid models developed in this chapter are designed in Simulink. However the total end-to-end system is rather complicated and several MATLAB functions are combined with Simulink for obtaining efficient and accurate solutions. The cross-traffic flow traces are extracted from a similar network configuration as the one used in the thesis by S.Doddi in [50].

E. Chapter Summary

This chapter presents the models of the subsystems that make-up the total end-to-end transport system. Models are constructed in the continuous-time. These models are later used to simulate the system under investigation as well as for designing the flow controllers.

CHAPTER IV

CONTROLLER DEVELOPMENT

The end-to-end system is modeled in Chapter III. Now, compensation methods for the system are outlined. At first, the objectives of the control problem are discussed and desired performance requirements are defined. Various available control strategies are explored. Control structures are synthesized based on the desired controller performance objectives. System identification techniques used in predictive schemes are described and unconstrained model predictive control techniques are developed.

A. Desired Controller Performance Criteria - Controller Objectives

Several performance criteria are now established as the effectiveness of the application level media flow controllers will be determined based on these criteria. However, as it was mentioned in Chapter II, achieving all of these objectives is a rather complicated task.

Playback disruption at the destination must be prevented. The destination buffer must be kept within certain bounds to prevent disruptions. However, this objective can also be achieved if the playback dead-time is increased. If the playback start time is delayed by increasing the desired setup buffer level x_{on}, then the buffer will most likely never empty and playback with no disruption is achieved. The tradeoff will be the dead-time increase, which is what is typically done in non real-time media applications.

Flow loss should be minimized. Ideally, a controller should prevent losses all together. However, this is not always possible. Any controller able to achieve some decrease in the amount of flow loss compared to the uncontrolled case, can be considered as beneficial. This also implies that congestion control is aided by the application

level controller. In order to achieve flow loss decrease, the send rate should be lower during periods of high delay, when the network is congested and when normally losses occur. As a result, lower send rates and in turn, less flow will be in transit lowering the probability of flow loss. Also, considering that the losses are lower, the flow throughput achieved is higher.

Playback start time at the destination should be minimized to decrease the end-to-end dead-time. Nevertheless, the dead-time can never be less or even equal to the dead-time in the open-loop case if no losses occur during that period of time. As explained earlier, in order to implement the control, buffering in the source as well as minimal buffering at the destination are necessary. Therefore further increase of the dead-time is unavoidable. However, if a controller achieves loss reduction during the dead-time, the throughput achieved during the same time will be higher. Therefore the destination buffer can reach the initial desired level faster and dead-time may be reduced.

B. Available Control Strategies

Before beginning the controller synthesis, consider some of the basic control structures that are available for this problem. All the control structures must be compared to the uncontrolled case when UDP is the transport protocol at the transport layer and no feedback is used at the application layer. Depending on the control methods, any number of signals may be useful for control purposes. The destination buffer level, $x(t)$, and the forward end-to-end transport delay, $\tau(t)$, as well as the accumulation, $a(t)$, are the main signals used in this work. Also, the send rate, $u(t)$, may be a required measurement. Controllers based on combinations of these signals are developed.

1. Reactive Feedback Compensation

The difference between the system output and the reference input, which is defined as the control error in feedback control, can be used as an input for the controller. In the case where end-to-end delay changes fast, reactive control is not reliable. If a feedback signal indicates that delay is high, and the delay changes fast, then by the time the controller reacts, the condition of the network will change. However, in this work as will be demonstrated in the simulation results the delays do not change fast. Therefore, a reactive controller can perform well as will be shown in the next chapter. The traditional PID controller is a reactive feedback controller. Again, reconsider that it is assumed that the feedback delay is negligible compared to forward delay and therefore the feedback signal at the current time is used.

2. Predictive Control Laws

If the output can be predicted, then the predictions can be used in the control structure to provide a predictive control scheme. The difference between predictive and reactive control, is that the latter reacts based on the last information available whereas the former reacts based on anticipated future information. Whether or not predictive can perform better is now the question. If the prediction is inaccurate, even the uncontrolled case outperforms predictive control. A controller can be predictive even if its structure is reactive, if the actual feedback signal is replaced by an estimate of the future signal. For the current development, it is assumed that prediction of the maximum delay is available. A model predictive controller can also be a likely solution to the problem. Linear empirical models can be used to develop multi-step predictors. The multi-step output predictors, can later be used to compute an objective function. The minimization of the objective function with respect to the control effort, leads

to the control law which is a vector of future control inputs.

3. Linear and Nonlinear Control Laws

Both linear and non-linear control laws may be used in both predictive and reactive feedback controls. However the simplistic structure of the linear controller is unable to provide the desired compensation as the linear controller does not consider the non-linear dynamics of the network. Nevertheless, the classic PID controller is also used and the results are presented in the next chapter. Linear reactive control can be used for regulating the destination buffer level.

C. Controller Development

Each controller is separately defined. Predictive, reactive and linear features are combined to synthesize the controllers. Various feedback signals are used to provide a robust and effective control solution for real-time Internet media applications.

1. Linear Controller (Controller 1)

The linear controllers' potential when used for compensation in highly non-linear systems is limited. As shown by Mangan in [29], sometimes such controllers perform worse than the uncontrolled cases.

The structure of the well known PID controller is familiar. It can be used for reference tracking and therefore it would be suitable for destination buffer regulation. The reference input is the desired buffer level x_d. The difference between desired and the actual buffer level or the level error is defined as

$$e(t) = x_d - x(t). \tag{4.1}$$

Hence, a term equal to the desired playback rate and a proportional, an integral and a differential feedback term are used to achieve this buffer regulation, as follows:

$$u_{id}(t) = d_{desired}(t) + k_P e(t) + k_I \int_{t_0}^{t} e(t)dt + k_D \frac{de(t)}{dt}. \tag{4.2}$$

The desired bit rate is $d_{desired}(t)$, the proportional, integral and differential constants are k_P, k_I and k_D respectively, the desired buffer level is x_d and the actual buffer level is $x(t)$.

2. Nonlinear Compensation Based on End-to-End Delay Measurement Feedback. (Controller 2)

The controller developed in this section has reactive structure as well. However if instead of a present feedback, an estimate of the future signal is used to calculate the control effort, the controller can be considered as predictive. The two cases are treated separately.

a. Reactive Nonlinear Compensation Based on End-to-End Delay

In order to reduce the losses, one needs to observe the time when they occur. In Chapter III the relation between delay and the network router's queue level was discussed. Losses occur when the router buffer level is close to the capacity of the buffer. Therefore when the delay is high it is likely that flow loss will occur. As a result send rates should be decreased when delay is high and increased when delay is low. In addition to that, developing a control law based on the general concept outlined above, aids in congestion control efforts as well. It is assumed that the present delay is a signal known to the controller. The inverse control law is of the form,

$$u_{inverse}(t) = k_0 \left(\frac{1}{\tau(t) - k_1} \right), \tag{4.3}$$

where k_0 and k_1 are control law constants. The constant k_1 is chosen to be near, but less than the lowest expected time delay which is the propagation delay. When the measured delay approaches k_1, the control effort becomes very large. The constant k_0 is chosen to make the control effort close to the desired playback rate. The term defined in equation (4.3) does not take into account the destination buffer level. Since the goal is to achieve reduction in losses while guaranteeing a continuous presentation rate from the destination buffer, a proportional term is added as follows,

$$u_{proportional}(t) = k_P \left(x_d - x(t) \right). \tag{4.4}$$

The integral of the total control effort for a sufficient length of time should be near that of the integral of the desired play rate over the same time period. Therefore, in order to keep the control effort close to the desired playback rate, integral control based on the integral of the difference between the send rate and the constant desired playback rate is employed. The integral control law is

$$u_{integral}(t) = k_I \int_{t_0}^{t} \left(d_{desired} - u(t) \right) dt, \tag{4.5}$$

where k_I is the integral control constant and $d_{desired}$ is the desired playback rate, as mentioned earlier.

If the control effort increases, especially at times of low end-to-end delay, the integral controller reduces the signal keeping it closed to the desired playback rate at the destination.

Adding the three terms together, the total control law becomes

$$u_{id}(t) = k_0 \left(\frac{1}{\tau(t) - k_1} \right) + k_P \left(x_d - x(t) \right) + k_I \int_{t_0}^{t} \left(d_{desired} - u(t) \right) dt. \tag{4.6}$$

b. Predictive Nonlinear Compensation Based on End-to-End Delay

The structure of the previous controller, is maintained for the controller in the present section also. However, if predictive information is used to determine the control effort then the controller is no longer considered as reactive. It is assumed that accurate delay prediction is available. Then, for this controller the estimation of the maximum delay is used. The same control effort as the one in equation (4.6) is applied. Let the maximum delay be τ_{max}. Then, when the actual end-to-end delay is in the region of $0.9\tau_{max}$ and τ_{max}, the control effort is reduced. The above, can be mathematically expressed as

$$u_{id}(t) = \begin{cases} k_0 \left(\frac{1}{\tau(t) - k_1} \right) + k_P \left(x_d - x(t) \right) + k_I \int_{t_0}^{t} \left(d_{desired} - u(t) \right) dt, & \text{if } \tau(t) < 0.9\tau_{max}, \\ k_2 d_{desired}, & \text{if } \tau(t) \geq 0.9\tau_{max}. \end{cases} \tag{4.7}$$

The constant k_2 may vary from 0 to 1. However, the best value for the cases studied has found to be 0.6.

The goal of this controller is flow loss reduction while maintaining a desired destination buffer level. Completely shutting off the controller, which occurs with $k_2 = 0$, is not desirable, since shutting off the control effort at the source for a short time would eventually lower and most likely empty the destination buffer. On the other hand, if the constant is selected to be 1 or even 0.9, then there would be no remarkable benefit, compared to the uncontrolled case where the send rate is maintained at the constant, $s(t) = d_{desired}$.

3. Nonlinear Compensation Based on Accumulation Measurement Feedback (Controller 3)

Delay predictions may not always be available. For that reason, an alternative solution is sought for. While maintaining the structure of Controller 2, an alternative signal is used instead of delay. However, the new signal can also be used in combination with delay signal. As shown in this section, delay and accumulation can be used in the same control equation.

a. Reactive Nonlinear Compensation Based on Accumulation

Consider that data is transferred from source to destination over a lossless network. The sending rate and arrival rate are given by $u(t)$ and $c(t)$, respectively. Then the accumulation at any time instant is given by

$$\alpha(t) = \int_{t_0}^{t} (u(t) - c(t))\, dt. \tag{4.8}$$

Equation (4.8), simply says that the difference of the cumulative sending rate and arrival rate gives the flow that is accumulated in the network queue or is in transit. When accumulation is high, this is an indication of congestion in the network. If the network is not assumed to be lossless, then equation (4.8) becomes

$$\lambda(t) = \alpha(t) + l(t) = \int_{t_0}^{t} (u(t) - c(t))\, dt, \tag{4.9}$$

where $\lambda(t)$ is the integral of the flow mismatch, a measured signal.

In Chapter II it was mentioned that $\frac{dl(t)}{dt}$ represents the loss rate signal. The integral of this represents the cumulative loss signal $l(t)$. In the case of equation (4.9) the signal $\lambda(t)$, has an increasing trend due to the term $l(t)$. On the other hand, the signal needed for control purposes is the accumulation $a(t)$, since it is the signal

which provides the controller an indication of congestion. Therefore, the de-trended signal can approximate the real accumulation signal and can be used as a feedback to the controller. A control law based on a form of the inverse of the accumulation signal is used.

In order to de-trend $\lambda(t)$, we calculate the moving average of the signal. The following equation applies

$$\mu(t) = \frac{\int_{t-\Delta}^{t} \lambda(s)ds}{\Delta}, \tag{4.10}$$

where Δ is a constant period of time called moving window. The accumulation is approximated by

$$\alpha(t) \approx \lambda(t) - \mu(t). \tag{4.11}$$

The inverse control law is chosen as the inverse square function

$$u_{inverse}(t) = k_0 \left(\frac{1}{\alpha(t) + k_1} \right)^2. \tag{4.12}$$

The rest of the terms are the same as those of Controller 2. Therefore, the following control law is used

$$u_{id}(t) = k_0 \left(\frac{1}{\alpha(t) + k_1} \right)^2 + k_P \left(x_d - x(t) \right) + k_I \int_{t_0}^{t} \left(d_{desired} - u(t) \right) dt. \tag{4.13}$$

The inverse accumulation term has the same objective as the inverse delay term of Controller 2.

b. Predictive Nonlinear Compensation Based on Accumulation and Delay

Similarly as Controller 2, Controller 3 can be implemented with delay prediction. In this case it is assumed that delay measurements and prediction are available. In this case, Controller 2 could be used as well. However, using accumulation and delay in the same equation is another option which is explored. The predictive version of the Controller 3 can be mathematically expressed as,

$$u_{id}(t) = \begin{cases} k_0 \left(\frac{1}{\alpha(t)+k_1}\right)^2 + k_P \left(x_d - x(t)\right) + k_I \int_{t_0}^{t} \left(d_{desired} - u(t)\right) dt, & \text{if } \tau(t) < 0.9\tau_{max}, \\ k_2 d_{desired}, & \text{if } \tau(t) \geq 0.9\tau_{max}. \end{cases} \tag{4.14}$$

The constant k_2, is varied from 0 to 1 as in the case of Controller 2. The best value has been found to be 0.66.

4. Compensation Based on Model Predictive Control Techniques (Controller 4)

An unconstrained model predictive control (MPC) law is derived using least-squares technique over a finite receding horizon. The system model is in the continuous time and in order to apply the MPC, empirical linear models are developed. The system input is the send rate, $u(t)$, whereas the output $y(t)$, can be selected to be any destination signal. In this thesis the MPC at the is investigated by selecting the output to be the destination buffer level. Keeping the desired output constant should prevent playback disruptions. This will have no effect to the losses as it will be shown later. For this reason the signal $\lambda(t)$, which include the flow losses is attempted as well.

There are two fundamental steps involved in developing an MPC, as follows:

1. Identification of the system.

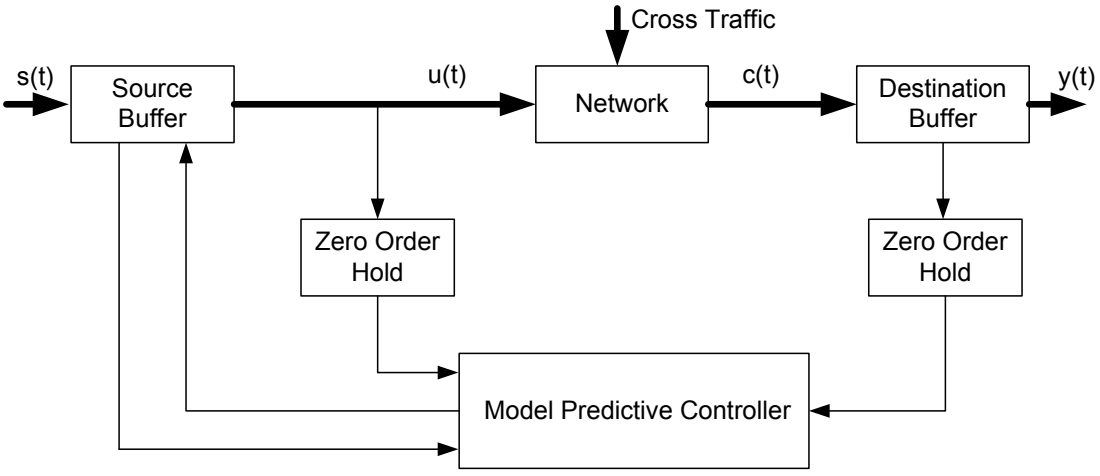

Fig. 7. A Block Diagram of the End-to-End System with Model Predictive Control.

2. Use of the identified model to compute control actions.

An Auto-Regressive with eXogenous input (ARX) model is used in this thesis for identification(ID) purposes. This model is used for both system ID and controller design. The parameters of the identified model are used to compute the predictive control law. The system with the predictive controller is shown in Figure 7.

At first, the uncontrolled system is simulated. Measurements of input and output are used to identify the single-input single-output system (SISO). Consider that the input is the send rate $u(t)$ and the output the destination buffer level $x(t)$. The model is in the continuous time, but discrete time histories of the input and output parameters should be used. For that reason a Zero Order Hold(ZOH) is used which samples the investigated variables at discrete time instants with sapling period T_s. Let the output vector be $\mathbf{y}$ and the send rate vector $\mathbf{u}$. The relationship between the discretized inputs and the outputs, can be described by the following one-step ARX predictor which has the form

$$\hat{y}(k+1/k) = a_1 y(k) + a_2 y(k-1) + \ldots + a_{na} y(k+1-na)$$
$$+b_1 u(k-d+1) + b_2 u(k-d) + \ldots + b_{nb} u(k-d-nb+2), \tag{4.15}$$

where k is the specific sampling instant number, na is the number of poles, $nb-1$ is the number of zeros and d is the system dead-time. The coefficients a_i and b_i appearing in equation (4.15) are the unknown ARX parameters. The first of the two major computational steps is the system identification. That is, to estimate the unknown parameters of the ARX model. The step of identification is easily implemented in MATLAB code. The discrete time input and output measurements are inserted in equation (4.15) and the parameters are identified.

In order to design the MPC controller, multi-step output prediction equations must be derived. Each prediction step is of T_s seconds. The one-step ahead prediction equation is obtained from Equation (4.15).

The 2-step ahead prediction equation, can be written as follows

$$\hat{y}(k+2/k) = a_1^{(2)} y(k) + a_2^{(2)} y(k-1) + \ldots + a_{na}^{(2)} y(k+1-na)$$
$$+b_1^{(2)} u(k-d+2) + \ldots b_2^{(2)} u(k-d+1) + \ldots + b_{nb+1}^{(2)} u(k-d-nb+2), \tag{4.16}$$

and so forth for additional step-ahead predictors.

Note that $a_i^{(n)}$ defines the i^{th} parameter of the n-step ahead prediction equation and it is not necessarily equal to a_i which is the i^{th} parameter of the one-step ahead prediction equation.

From the above equations the predicted output for the j^{th} step-ahead prediction can be written in the following form

$$\hat{y}(k+j/k) = a_1^{(j)}y(k) + a_2^{(j)}y(k-1) + \ldots + a_{na}^{(j)}y(k+1-na) + b_1^{(j)}u(k-d+j) + \ldots + b_{j-1}^{(j)}u(k-d+2) + b_j^{(j)}u(k-d+1) + \ldots + b_{j+nb-1}^{(j)}u(k-d-nb+2). \quad (4.17)$$

The MPC algorithm is based on output predictions over a finite horizon hc. Then the future control inputs can be determined over hc, which is the maximum prediction horizon that will be used in the multi-step prediction equation development. Letting j range from 1 to hc the resulting equations can be combined into a multi-step prediction equation of the following matrix form

$$\begin{bmatrix} \hat{y}(k+1/k) \\ \hat{y}(k+2/k) \\ \vdots \\ \hat{y}(k+hc/k) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & \ldots & a_{na-1} & a_{na} \\ a_1^{(2)} & a_2^{(2)} & \ldots & a_{na-1}^{(2)} & a_{na}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_1^{(hc)} & a_2^{(hc)} & \ldots & a_{na-1}^{(hc)} & a_{na}^{(hc)} \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-na+1) \end{bmatrix}$$
$$+ \begin{bmatrix} b_2 & b_3 & \ldots & b_{nb-1} & b_{nb} \\ b_3^{(2)} & b_4^{(2)} & \ldots & b_{nb}^{(2)} & b_{nb+1}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{hc+1}^{(hc)} & b_{hc+2}^{(hc)} & \ldots & b_{nb+hc-2}^{(hc)} & b_{nb+hc-1}^{(hc)} \end{bmatrix} \begin{bmatrix} u(k-d) \\ u(k-d-1) \\ \vdots \\ u(k-d-nb+2) \end{bmatrix}$$
$$+ \begin{bmatrix} b_1 & & & \\ b_2^{(2)} & b_1^{(2)} & & \\ \vdots & \vdots & \ddots & \\ b_{hc-1}^{(hc-1)} & b_{hc-2}^{(hc-1)} & \ldots & b_1^{(hc-1)} \\ b_{hc}^{(hc)} & b_{hc-1}^{(hc)} & \ldots & b_2^{(hc)} & b_1^{(hc)} \end{bmatrix} \begin{bmatrix} u(k-d+1) \\ u(k-d+2) \\ \vdots \\ u(k-d+hc) \end{bmatrix}. \quad (4.18)$$

The Equation (4.18) can also be written in the form of

$$\hat{\mathbf{y}}_{hc} = \mathbf{A}\mathbf{y}_p + \mathbf{B}\mathbf{u}_p + \mathbf{T}\mathbf{u}_{hc}, \tag{4.19}$$

where $\hat{\mathbf{y}}_{hc}$, is the vector of the future outputs of dimension $hc \times 1$, $\mathbf{A}$ a matrix with the coefficients of past output measurements, with dimension $hc \times na$, $\mathbf{y}_p$ the vector of past output measurements with dimension $na \times 1$, $\mathbf{B}$ the matrix with the coefficients of past input measurements with dimension $hc \times (nb-1)$, $\mathbf{u}_p$ is the vector of past input measurements with dimension $(nb-1) \times 1$, $\mathbf{T}$ is the square matrix of coefficients of the future control inputs of dimension $hc \times hc$ and $\mathbf{u}_{hc}$ the future control sequence of dimension $hc \times 1$.

Equation (4.18) indicates that, the first d outputs depend only on past samples as expected, since d is the system dead-time. For this reason, the first d equations can be ignored.

The goal of the MPC algorithm is to determine the set of future commands $\mathbf{u}_{hc}$, which are required to drive the predicted output $\hat{\mathbf{y}}_{hc}$ as close to a desired output $\mathbf{y}_d$ (setpoint) as possible, in a least-squares sense. If the output is set to be the destination buffer level $x(t)$, then the desired output would be a constant value equal to the desired destination buffer level. Keeping the destination buffer close to a constant value will eliminate any playback disruptions. The error function, that is the difference between the desired and the predicted output is defined as

$$\epsilon \ = \mathbf{y}_d - \hat{\mathbf{y}}_{hc}. \tag{4.20}$$

The goal of the MPC is then equivalent to minimizing the quadratic objective function defined as

$$J = \epsilon^T \mathbf{R} \epsilon + \mathbf{u}_{hc}^T \mathbf{Q} \mathbf{u}_{hc}, \tag{4.21}$$

where $\mathbf{Q}$ and $\mathbf{R}$, are weighting matrices on the control effort and the tracking error respectively. Minimizing equation (4.21) and solving with respect to $\mathbf{u}_{hc}$ yields

$$\mathbf{u}_{hc} = (\mathbf{T}^T \mathbf{R} \mathbf{T} + \mathbf{Q})^{-1} (\mathbf{T})^T \mathbf{R} (\mathbf{y}_d - \mathbf{A}\mathbf{y}_p - \mathbf{B}\mathbf{u}_p). \tag{4.22}$$

Equation (4.21) gives the control commands to be applied over the next hc controller time steps. However, in accordance with MPC, only the first of the control actions is implemented and the receding horizon is advanced to the next time step to recompute the control law.

The real benefit of this method is that any output signal can be related to the input rate. If it is decided that instead of the buffer level, a different output should be regulated then a new model would be identified and the procedure would be repeated.

D. Chapter Summary

In this chapter four major control options are evaluated: linear control, non-linear reactive feedback control using delay measurements as feedback, non-linear reactive feedback control based on accumulation feedback, non-linear predictive control and model predictive control. These options are discussed and expressed mathematically. The effectiveness of these control methods is examined in Chapter V.

CHAPTER V

SIMULATION RESULTS

Simulation results using fluid-flow models are presented in this chapter. The simulation results are obtained utilizing both MATLAB and Simulink. In [50] a similar network architecture was developed using ns-2. The cross-traffic traces used in [50] are used for the purposes of the current development, following appropriate scaling to fit present study. The effectiveness of various control algorithms is determined for the specific network conditions under different source send rates.

A. Network Architecture

The topology depicted in Figure 3 is modelled in Simulink. The cross-traffic is obtained from an imported ns-2 file from a simulation consisting of traffic generated by TCP and UDP sources with a ratio of one UDP source for every ten TCP sources.

It is important to keep in mind that the network used in these simulations is normalized and only used to demonstrate the concept feasibility. Therefore the units used in this research are called simply "units". The term "units" is used instead of packets or bytes. The rates are measured by units per second (ups). The two cross-traffic traces used in this research are shown in Figure 8.

The bottleneck link capacity C, is selected to be 2000 ups and the end-to-end propagation delay is chosen to be 500 ms (0.5 seconds). The baseline source send rate, which is the source buffer input rate is 150 ups, but the controllers are also tested under source rates of 120 ups and 180 ups.

The network architecture parameters are chosen to provide a reasonable delay trace. The cross-traffic traces are scaled to make the bandwidth required by the controlled flow a relatively small portion of the total bandwidth. However, the ratio

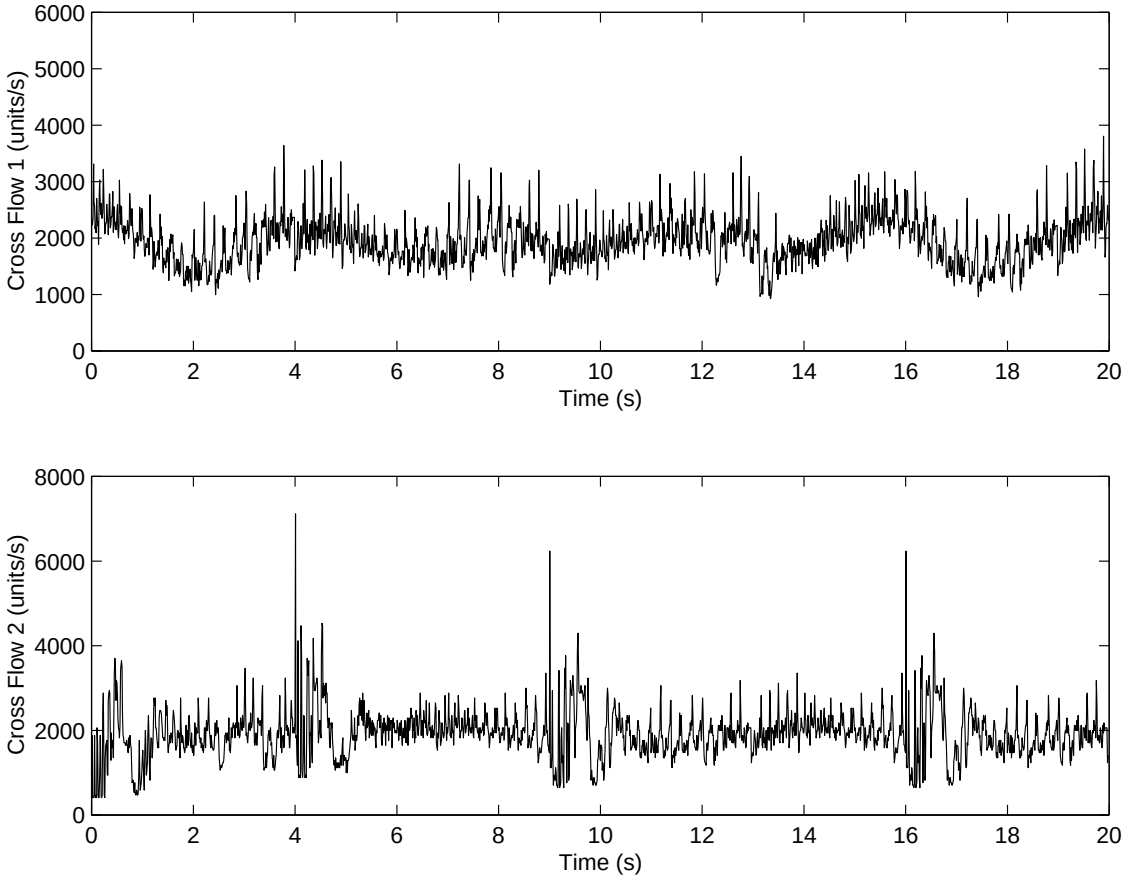

Fig. 8. Cross-Traffic Traces Used in MATLAB Simulations

of controlled flow to the cross-traffic flow is 1 to 10 and it is not realistically large.

In each of the simulations, the application send rate is started at time $t = 3$. The flow continues until near the end of the simulation, at about 15 seconds, and then stops to allow the destination buffer to run out. The simulation stops at $t = 20$ seconds. This simulates a media flow that enters into a network that is already operating and stops before the end of the simulation to allow the last segments of flow to arrive at the destination.

B. Effectiveness of the Controllers for Baseline Application Send Rate

In this section various simulations are demonstrated. The application send rate is kept at 150 ups which is the baseline. Each controller is separately investigated. Results for both cross-traffic traces are presented. At the beginning the open-loop simulation results are presented. The open-loop simulation will be used for comparing all of the other closed-loop simulations in order to determine the advantage of using a controller over the simple "no-control" scenario.

1. Open-Loop Simulations

Open-loop is the case when no control policy is implemented. Most real-time media applications use UDP and send packets at a constant rate without applying any congestion avoidance schemes or being concerned about improving the QoS. Consider that minimal buffering at the destination is also assumed. The x_{on} is low for the first simulation and does not prevent playback disruptions.

Consider the open-loop case as shown in Figures 9, 10 and 11. The destination buffer level, send rate, arrival rate and playback rate are shown in Figure 9. The end-to-end delay, loss rate and cumulative losses are shown in Figure 10. Finally, in

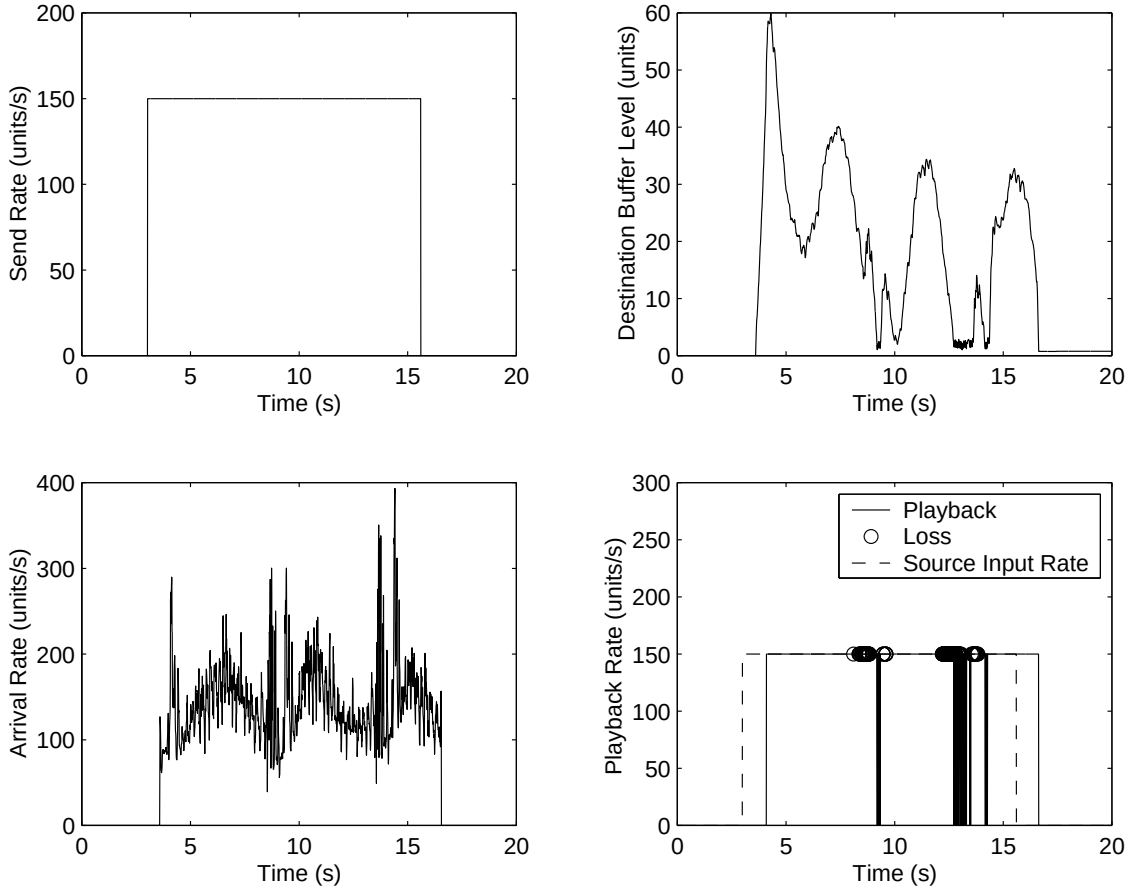

Fig. 9. Buffer Level and Flow Rates for Open-Loop Simulation Using Cross-Traffic 1.

Figure 11 the cumulative send rate and arrival rate, the difference of them $\lambda(t)$ and the accumulation $\alpha(t)$, are depicted. In Figure 9, the playback rate is compared to the application send rate. One can observe the total playback delay which is the time difference of the playback start time and the time when application starts sending data. The total playback delay, e.g. dead-time, is the sum of the end-to-end delay and the time of flow residence in the application buffers. In open-loop case no source buffering occurs, but in the control led cases the source buffering time is significant as will be shown shortly.

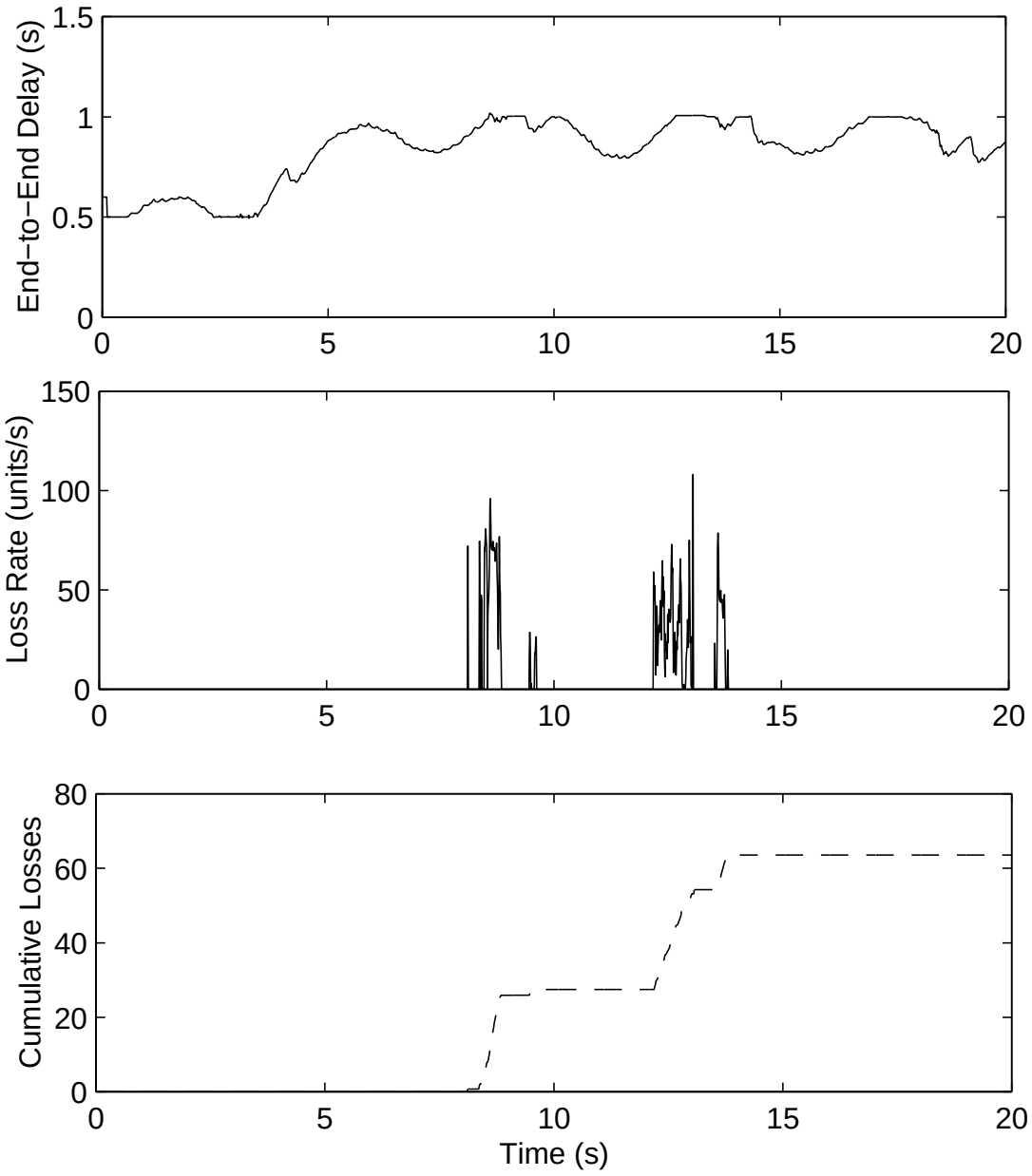

Fig. 10. End-to-End Delay and Losses for Open-Loop Simulation Using Cross-Traffic 1.

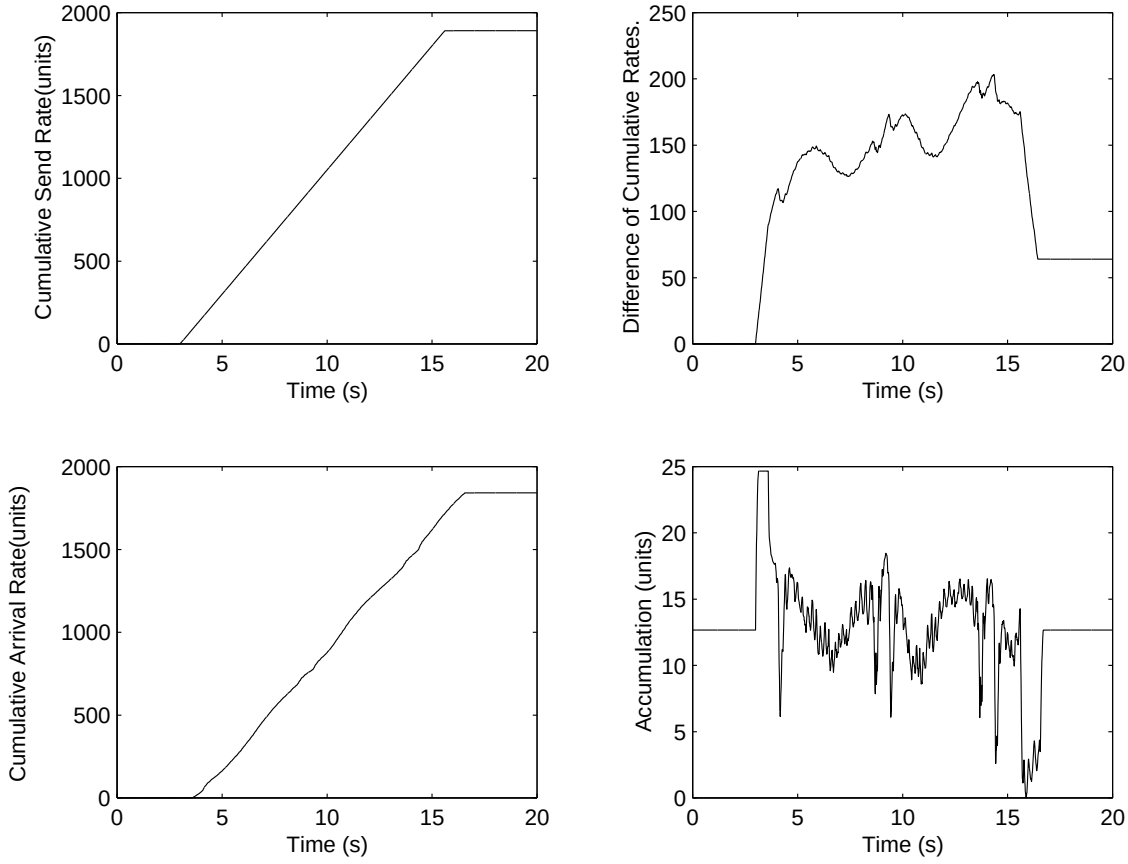

Fig. 11. Cumulative Flow Rates and Accumulation for Open-Loop Simulation Using Cross-Traffic 1.

High delays are symptoms of network congestion as shown in Figure 10. Therefore, at times of high delay, network losses will be high and the arrival rate will be lower due to the higher queues in the network. This is why playback at the destination is interrupted temporarily at $t = 8$ and $t = 12$ seconds. The plots of the loss rates and the cumulative losses are also shown in Figure 10. The loss rate plot shows when the losses occur. One can observe that flow is lost when delays are high. The cumulative loss plot shows the integral of the loss rate.

One of the control objectives is to eliminate disruptions. However one way to do this, is simply to increase the initial buffering at the destination. If x_{on} is increased sufficiently then no matter what the network delay is, the destination buffer will always have enough flow and playback will not be interrupted. The tradeoff will be the increase in the dead-time, due to the time needed for the flow to be stored. Figures 12, 13 and 14 show the various system parameters when destination initial buffering is increased using cross-traffic trace 1. The dead-time compared to the dead-time shown in Figure 9 is increased by 40%. Observe that there is no playback disruption in Figure 12. Figures 15, 16 and 17 show the system variables for the cross-traffic trace 2.

The previous case studies will be the basis for comparison for the controlled cases. Since playback disruptions can be prevented by simply buffering at the destination, then the question becomes what is the benefit of implementing control. If in both open-loop and controlled cases the disruptions can be avoided then the losses and the dead-time should be compared. In the controlled scenarios, the buffering occurs at the source instead of the destination. The more the initial source buffering is increased, the less likely is for the source buffer to empty. When the source buffer empties the controller can not implement the calculated control actions and the send rate is limited to the constant application send rate. On the other hand the initial source

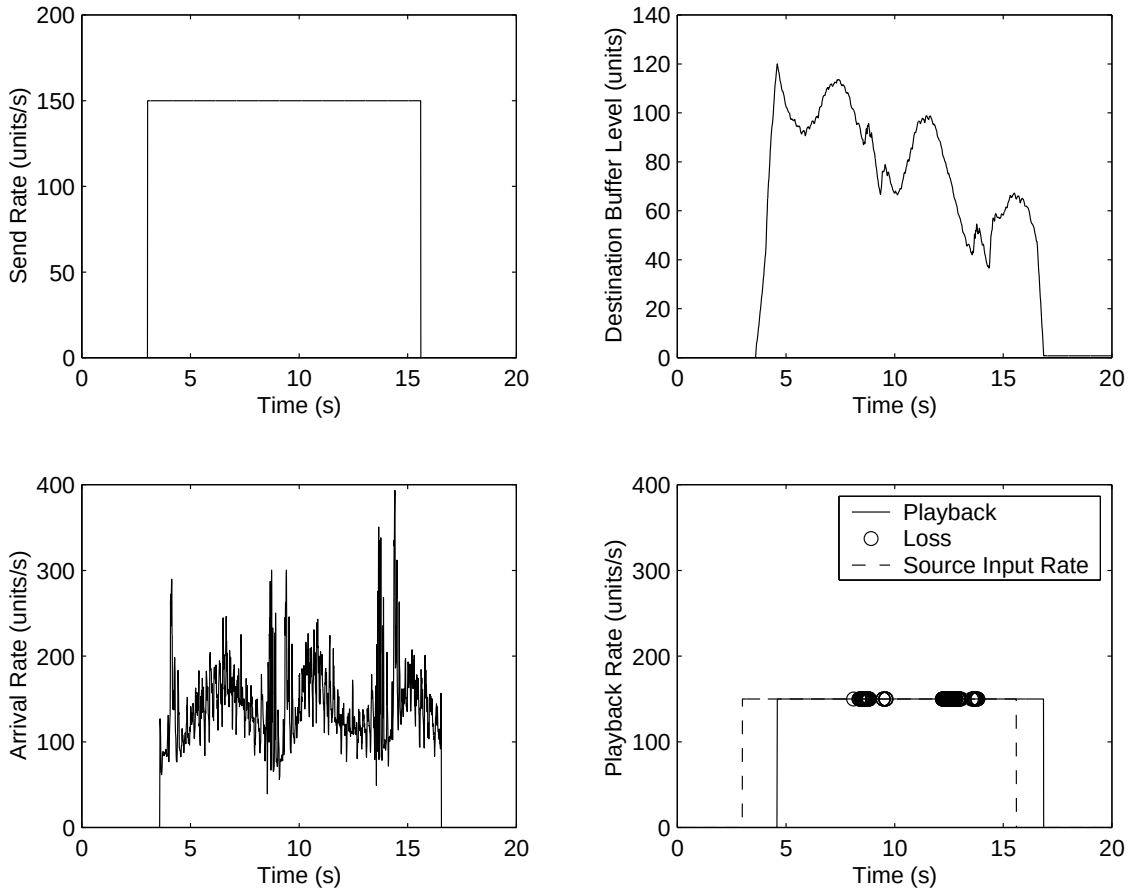

Fig. 12. Buffer Level and Flow Rates for Open-Loop Simulation Using Cross-Traffic 1 for Increased x_{on}.

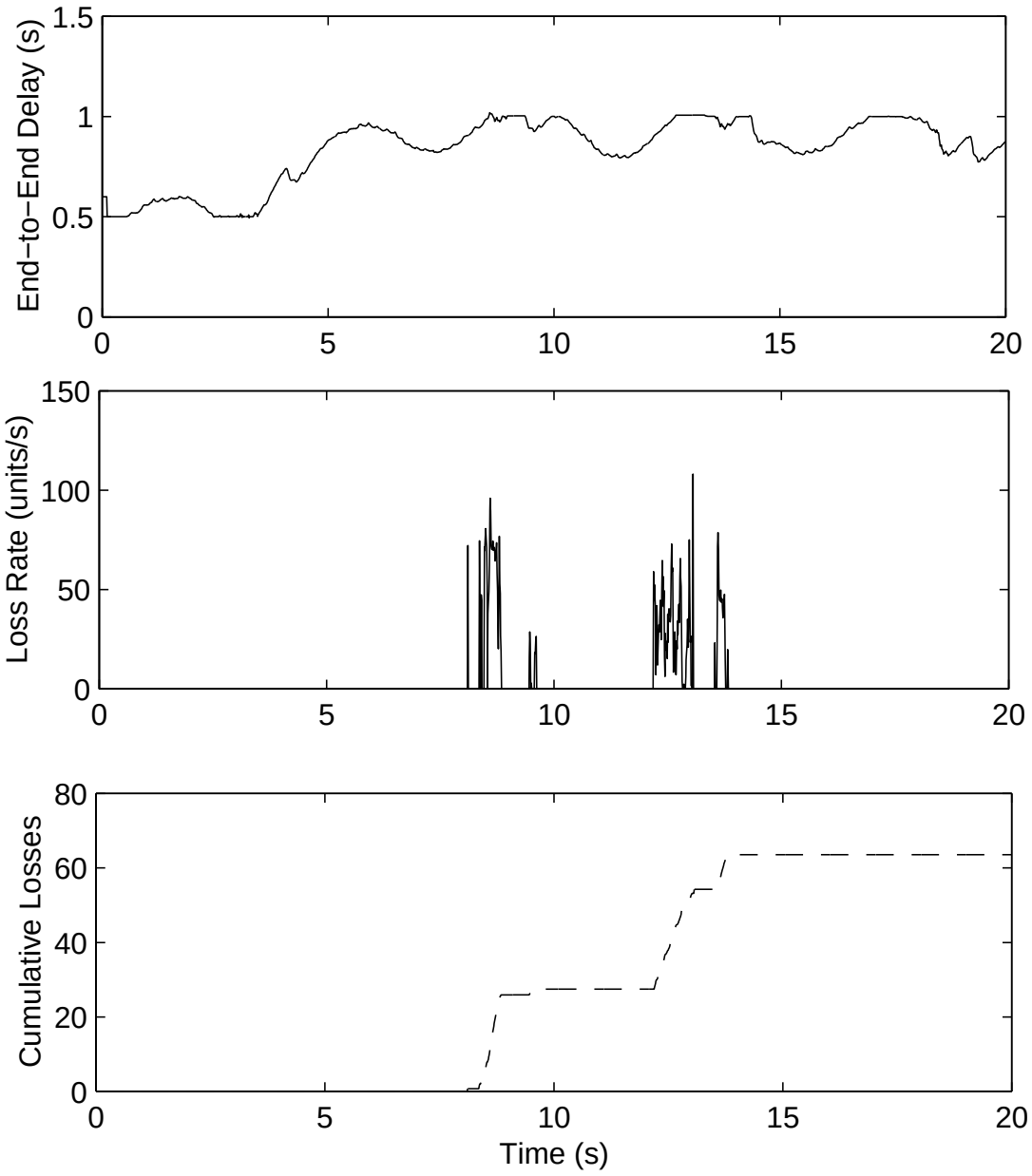

Fig. 13. End-to-End Delay and Losses for Open-Loop Simulation Using Cross-Traffic 1 for Increased x_{on}.

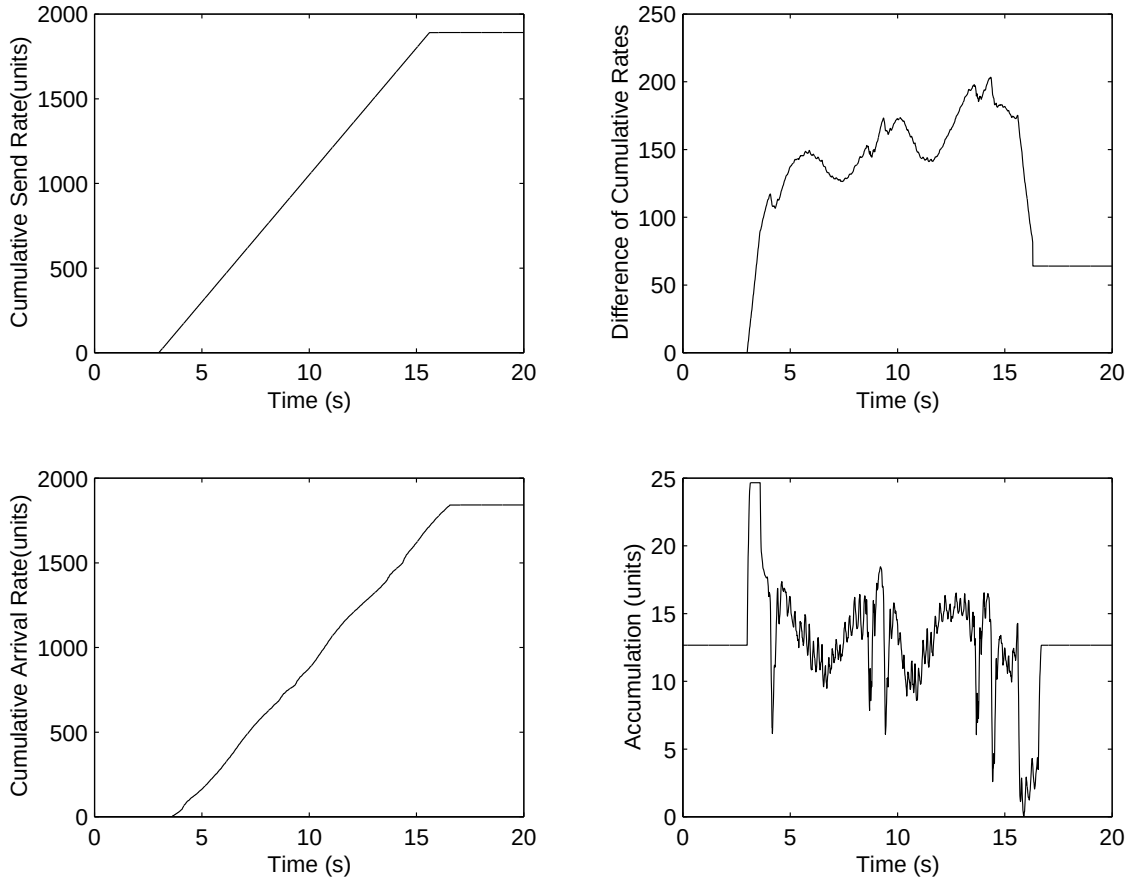

Fig. 14. Cumulative Flow Rates and Accumulation for Open-Loop Simulation Using Cross-Traffic 1 for Increased x_{on}.

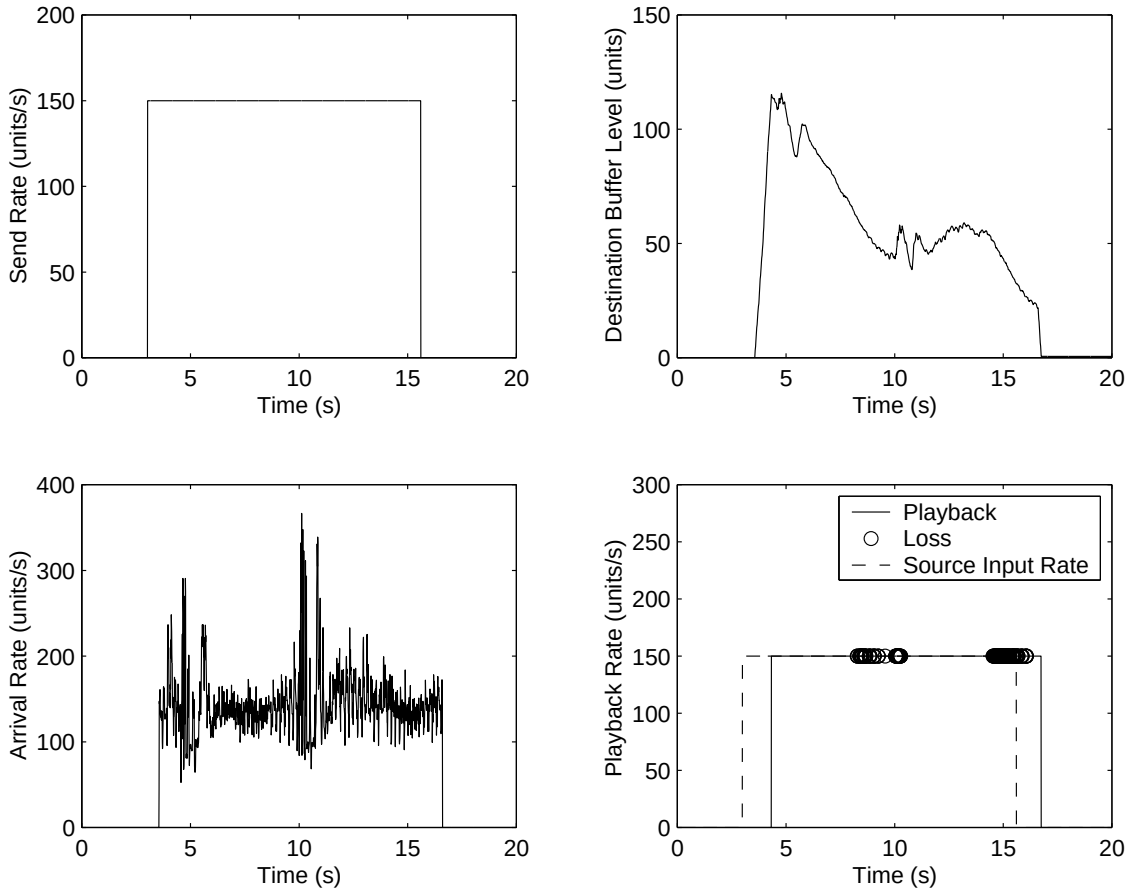

Fig. 15. Buffer Level and Flow Rates for Open-Loop Simulation Using Cross-Traffic 2 for Increased x_{on}.

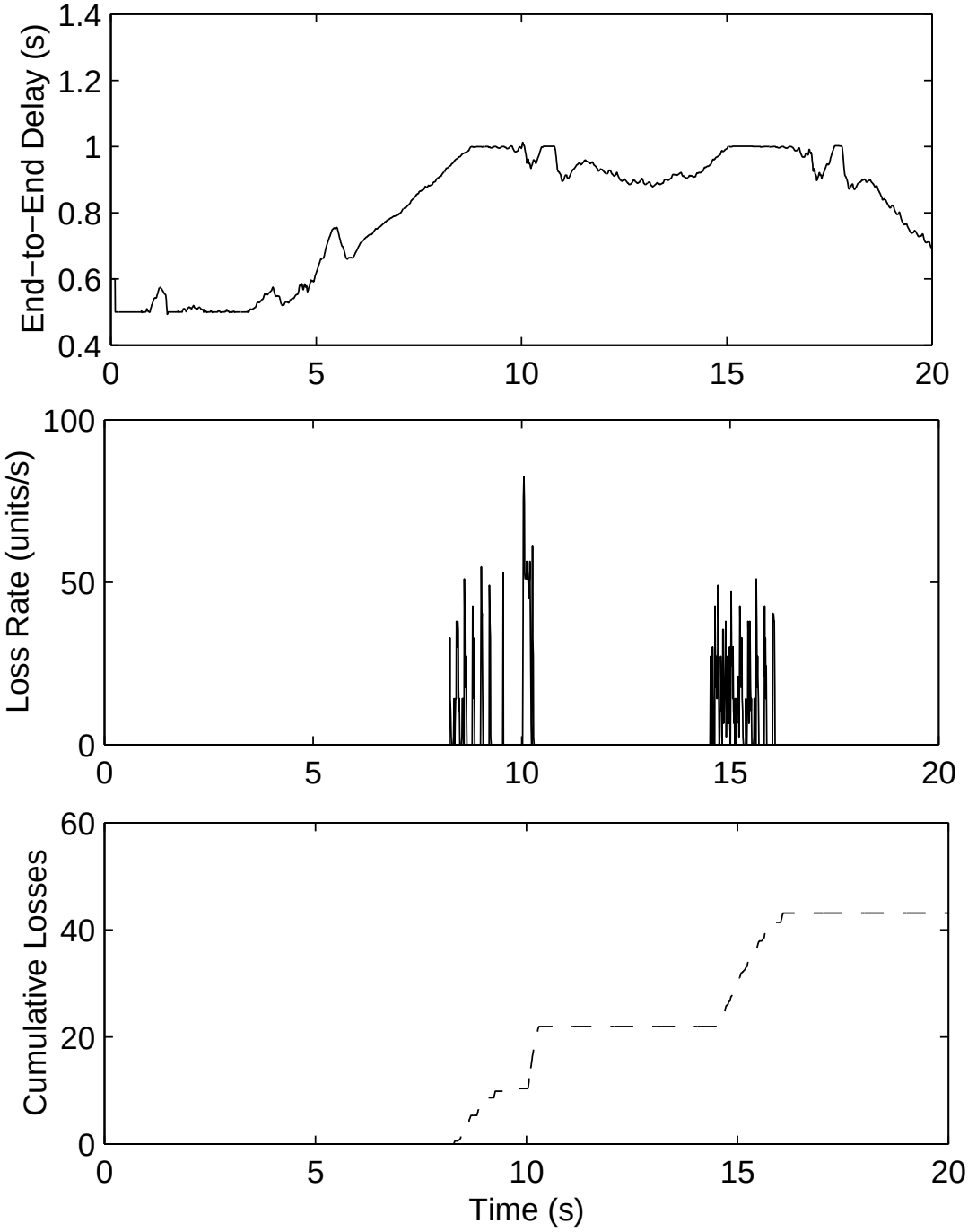

Fig. 16. End-to-End Delay and Losses for Open-Loop Simulation Using Cross-Traffic 2 for Increased x_{on}.

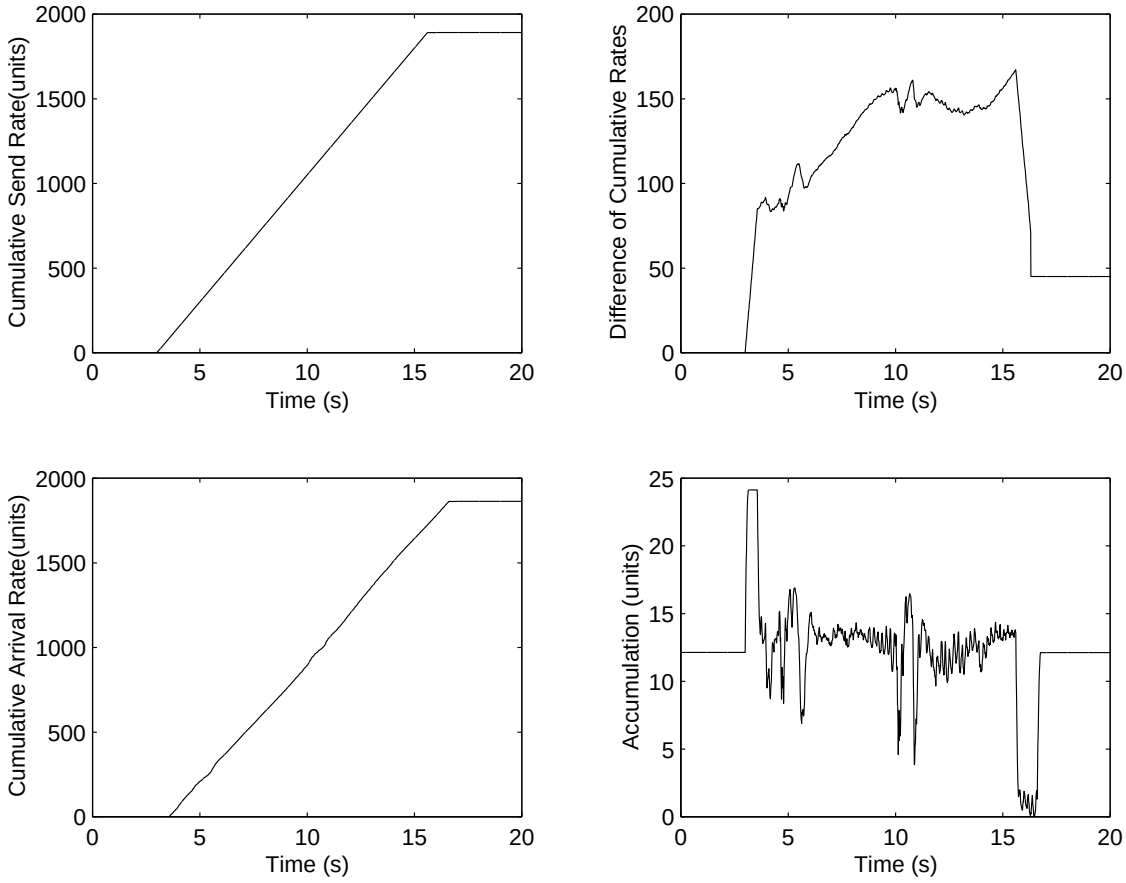

Fig. 17. Cumulative Flow Rates and Accumulation for Open-Loop Simulation Using Cross-Traffic 2 for Increased x_{on}.

Table I. Performance Parameters for the Open-Loop Case Using Different Cross-Traffic Traces.

Cross-Traffic	**Number of Unit Losses**	**Dead-Time**
Cross-Traffic 1	65	1.62 seconds
Cross-Traffic 2	44	1.40 seconds

buffering increase increases the system dead-time. This is clearly demonstrated in the following sections.

The Table I, summarizes the main performance parameters for the open-loop cases. The effectiveness of the controllers is determined based on these performance metrics. A beneficial controller should achieve a decrease on the number of losses while keeping the dead-time close to that shown in Table I. The dead-time in the control led cases can be less than that of the open-loop case only if losses are prevented during this time. In this case, the throughput would increase and the x_{on} would be reached faster, compared to the open-loop case.

2. Linear Control - Controller 1

Simulations of the linear Controller 1 are now presented. The objective of this classical PID controller was to regulate the buffer level in order to minimize disruptions. Although the controller achieves its objective in regulating the destination buffer, the losses and the dead-time are increased 7% and 11%, respectively as compared to the open-loop case. Figures 18 and 19 show the end-to-end parameters for Controller 1 simulation using cross-traffic 1.

The Figures 20 and 21 show the same parameters for cross-traffic 2. Although the objective of the controller is achieved, the losses and dead-time are increased by

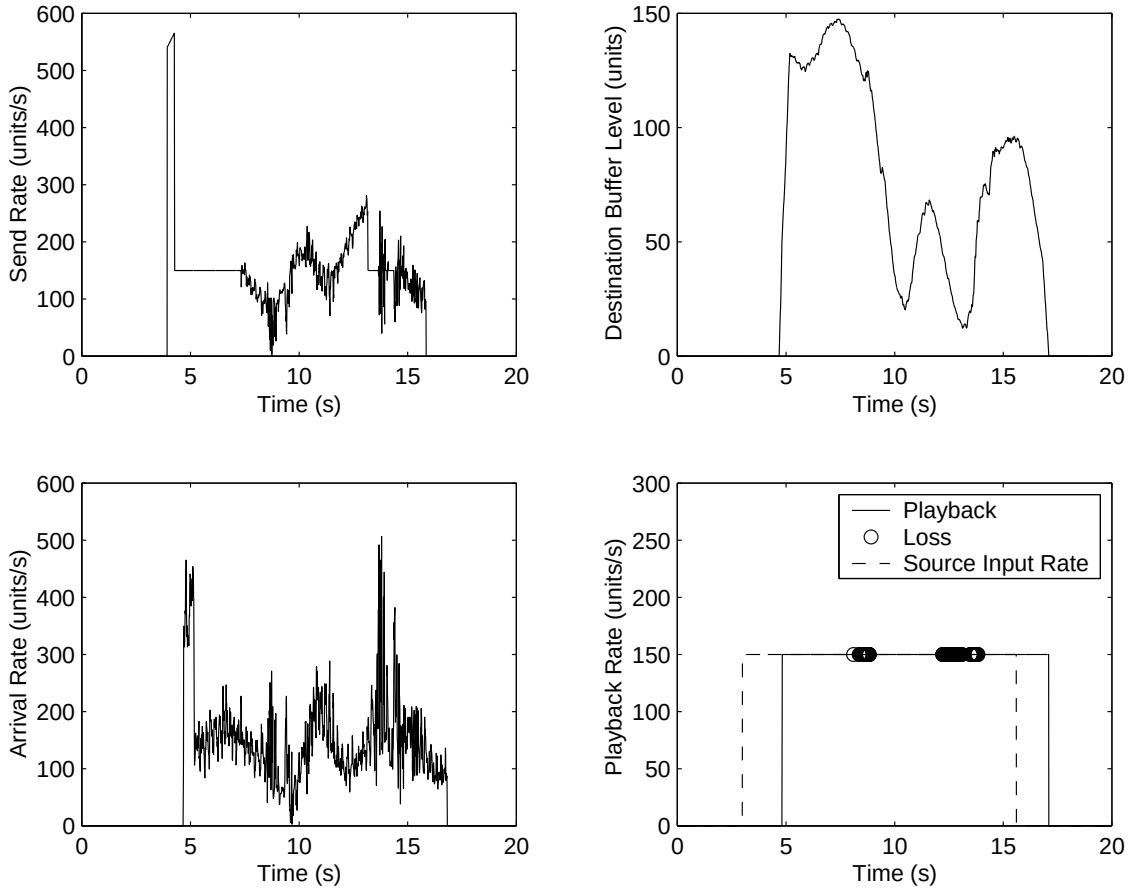

Fig. 18. Buffer Level and Flow Rates for Controller 1 Simulation Using Cross-Traffic 1.

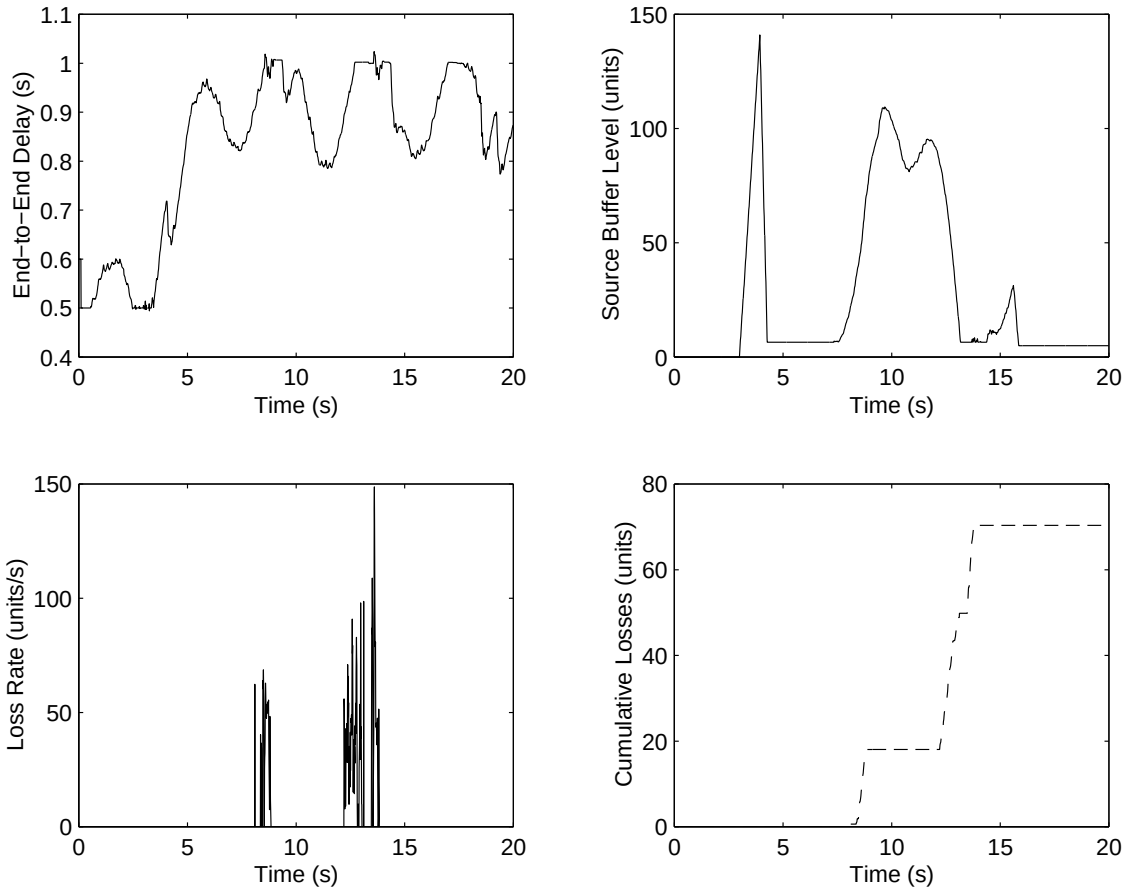

Fig. 19. End-to-End Delay, Source Buffer Level and Losses for Controller 1 Simulation Using Cross-Traffic 1.

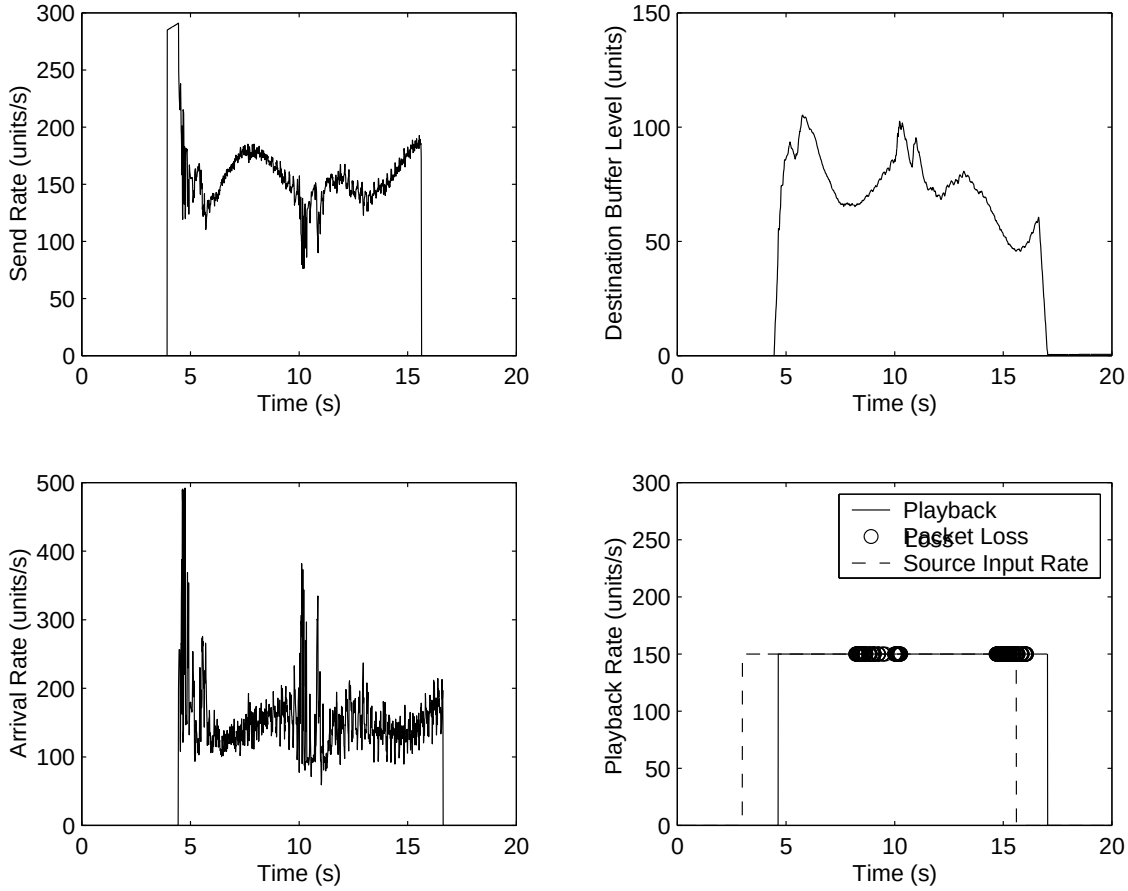

Fig. 20. Buffer Level and Flow Rates for Controller 1 Simulation Using Cross-Traffic 2.

15% and 10% respectively as compared to the open-loop case. This was expected as the controller was not designed to prevent losses. The single objective of regulating the buffer forced the controller to increase the rate when the buffer was low. Normally buffer level lowers when delay is high. And therefore loses increase as the send rate is increased at times of high delay. The simple structure of the linear Controller 1 has limited potential. No complicated objectives can be achieved without "violating" the linear structure.

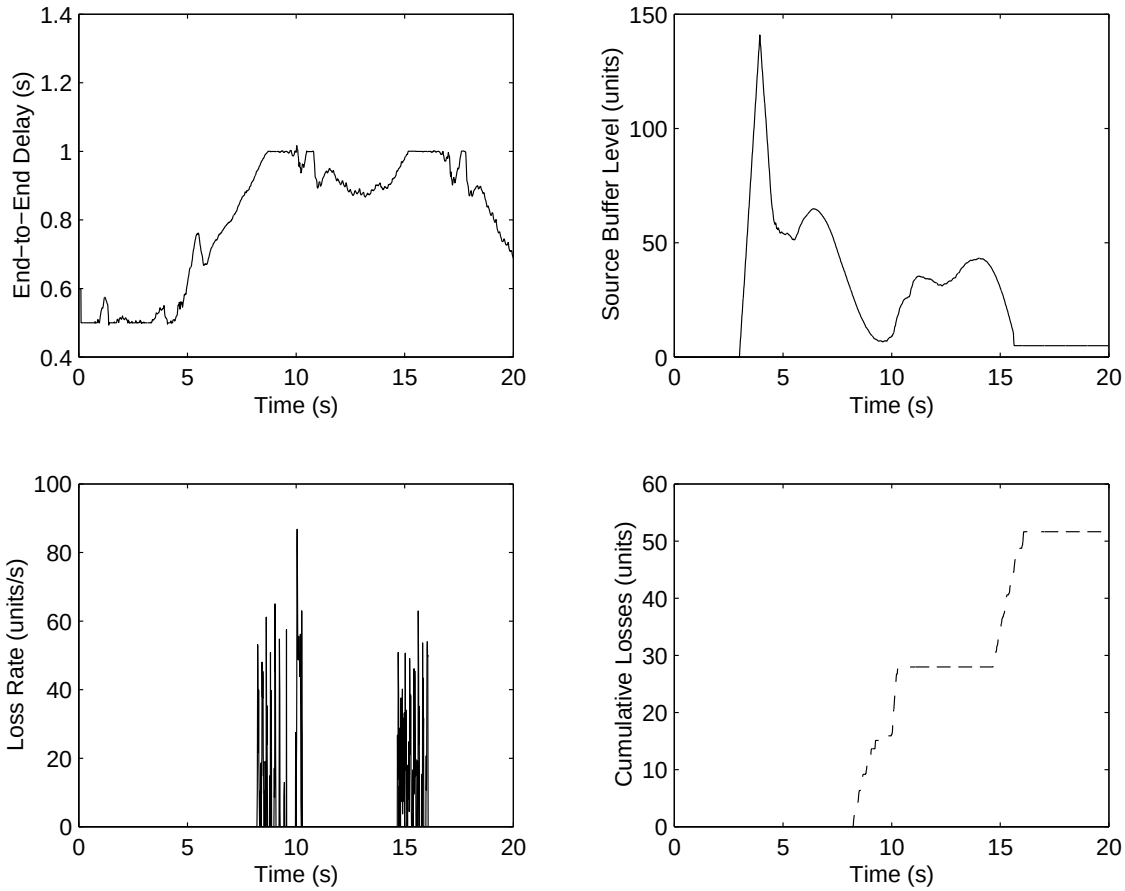

Fig. 21. End-to-End Delay, Source Buffer Level and Losses for Controller 1 Simulation Using Cross-Traffic 2.

3. Nonlinear Control - Controller 2

Controller 2 can be implemented either reactively or predictively. In this section both cases are presented. This controller uses the end-to-end delay in order to reduce the losses. The simulations show that this objective is achieved.

a. Reactive Implementation of Controller 2

The Figures 22 and 23 show the system response simulation results for this controller. The losses are reduced by 22% whereas the dead-time is increased by seconds 13% over the open-loop case.

In the case of cross-traffic 2 the same controller manages to decrease the losses by 9% although the dead-time is increased by 14% as one can see in Figures 24 and 25.

Controller 2 is based on the inverse of the delay. One can observe how the send rate is increased when the delay is low and it is decreased when the delay is high. The control in Figure 22 is initially high due to the integral term of the controller. The desired playback is constant while the send rate is zero due to the initial source buffering. Therefore the difference of them is higher initially. Also the destination buffer is empty and the proportional term is higher. For these reasons the controller tries to compensate minimizing the difference between desired and actual buffer level. Although the controller is reactive it manages to achieve loss reduction. Observe the delay plot of the system in Figures 23 and 25. It shows that the delays change slowly for both cross-traffic traces. During actual network operating conditions the delays change rapidly and the use of reactive control algorithms would not be of much use. However the simulations of this work prove that when the delays of the network change slowly, the use of reactive control can be beneficial. The following simulations

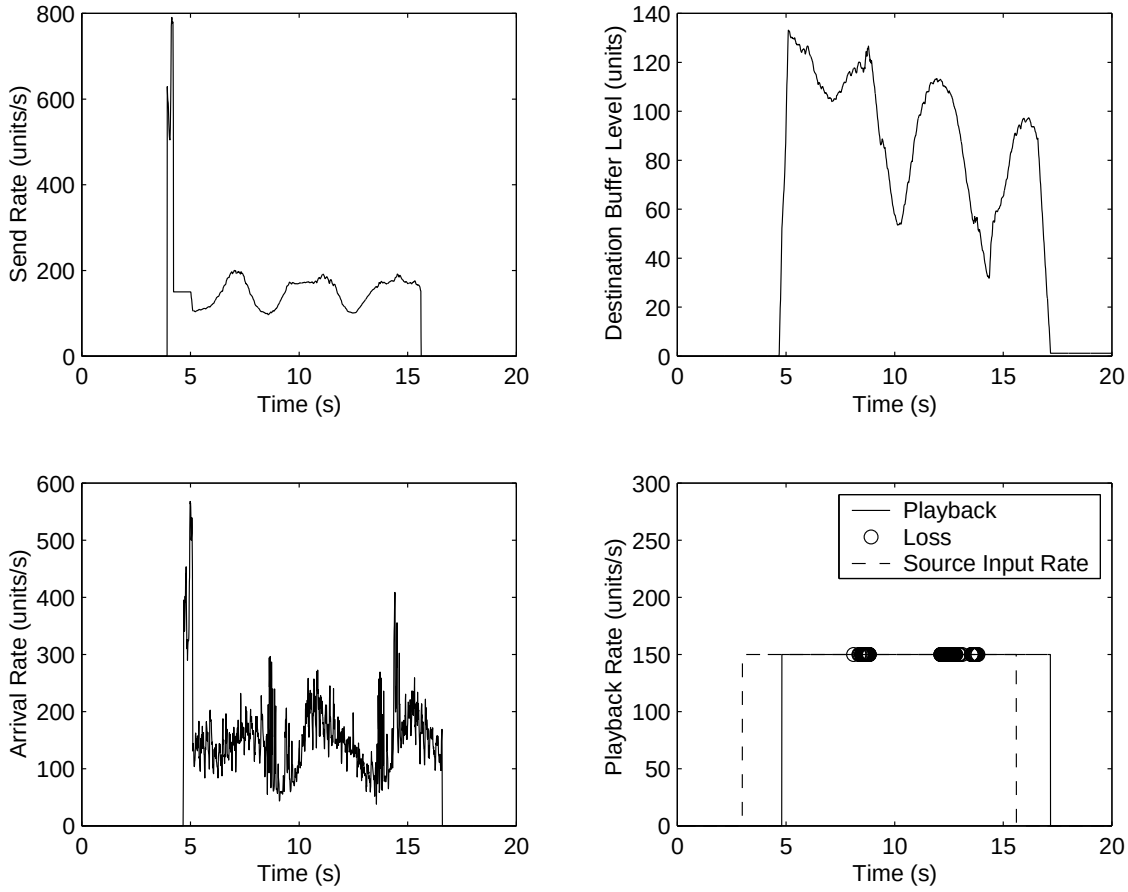

Fig. 22. Buffer Level and Flow Rates for Reactive Controller 2 Simulation Using Cross-Traffic 1.

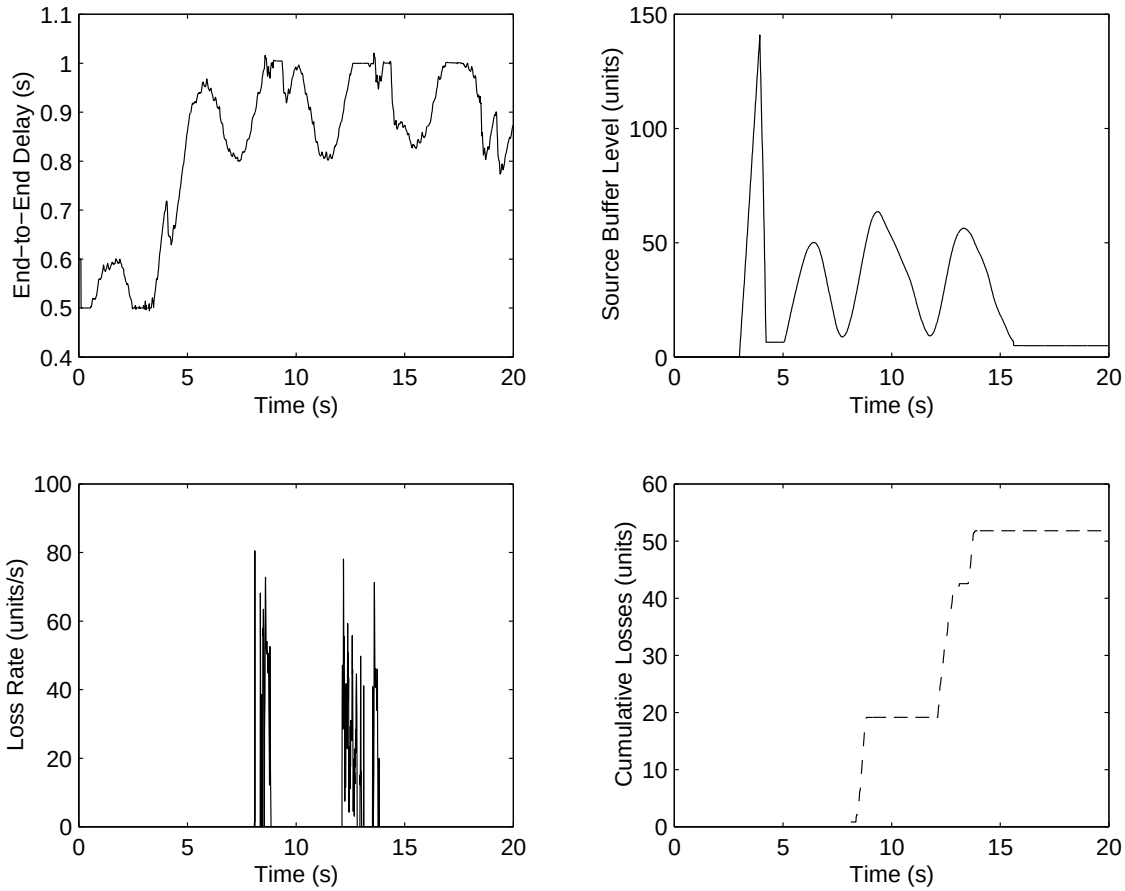

Fig. 23. End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 2 Simulation Using Cross-Traffic 1.

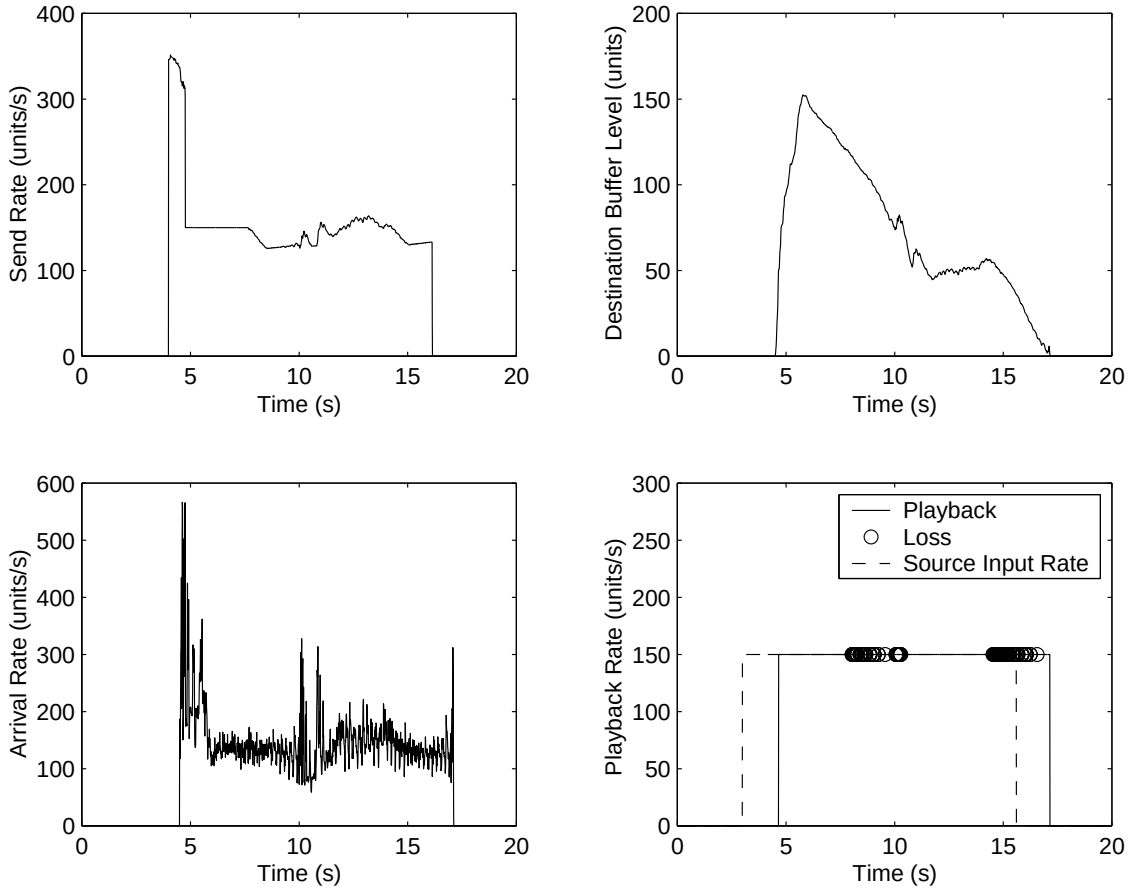

Fig. 24. Buffer Level and Flow Rates for Reactive Controller 2 Simulation Using Cross-Traffic 2.

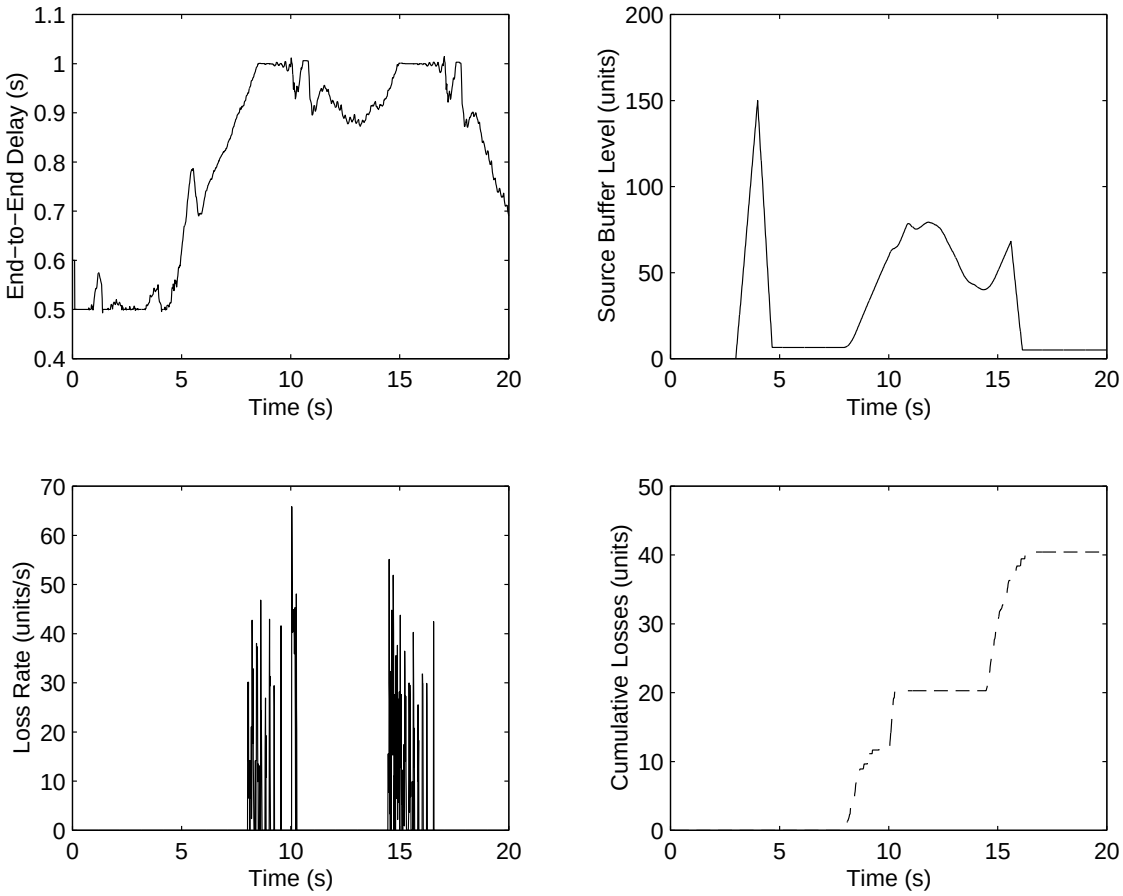

Fig. 25. End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 2 Simulation Using Cross-Traffic 2.

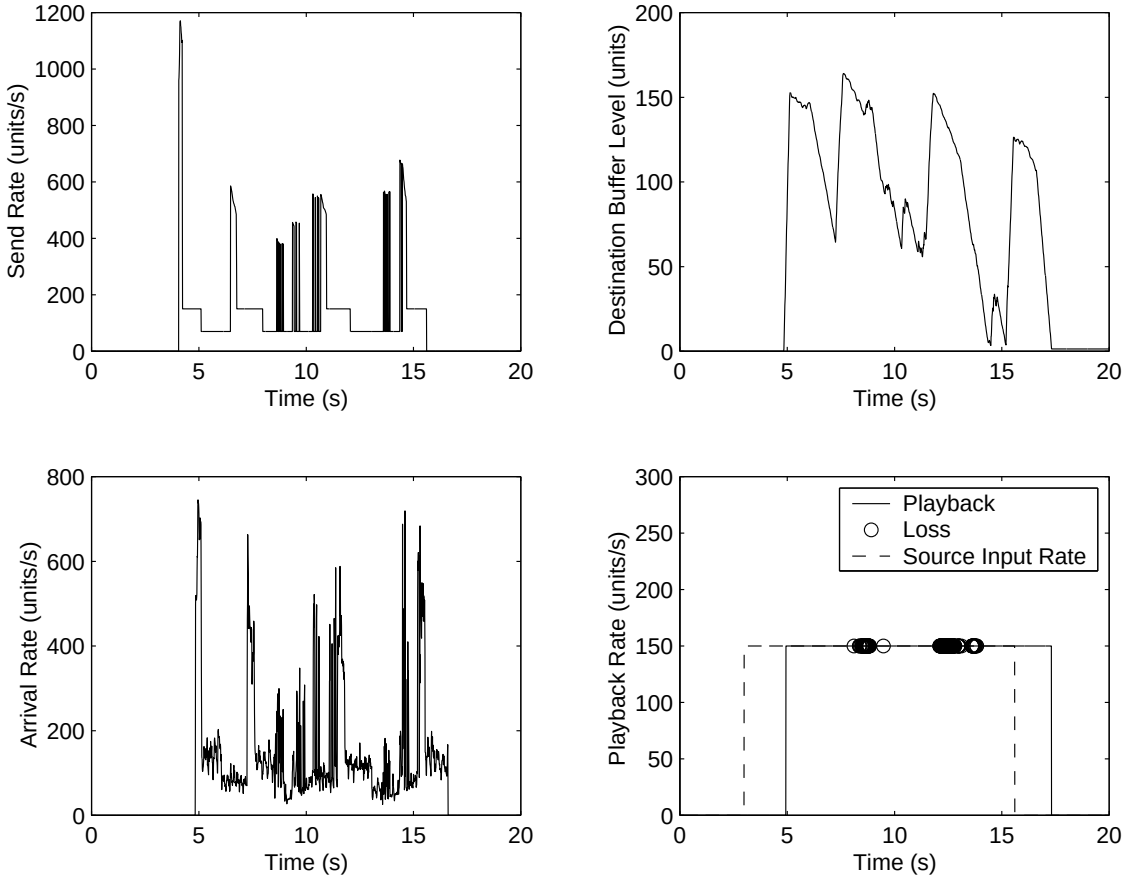

Fig. 26. Buffer Level and Flow Rates for Predictive Controller 2 Simulation Using Cross-Traffic 1.

with reactive controllers will also support this argument.

b. Predictive Implementation of Controller 2

If the controller uses an estimate of the maximum anticipated delay that will be experienced, then it can no longer be considered as reactive. In this case, the control effort is decreased at times when the end-to-end delay is above 90% of the maximum delay. Figures 26 and 27 show that the losses are decreased by 40% and the dead-time is increased by 18%, all compared to the uncontrolled case.

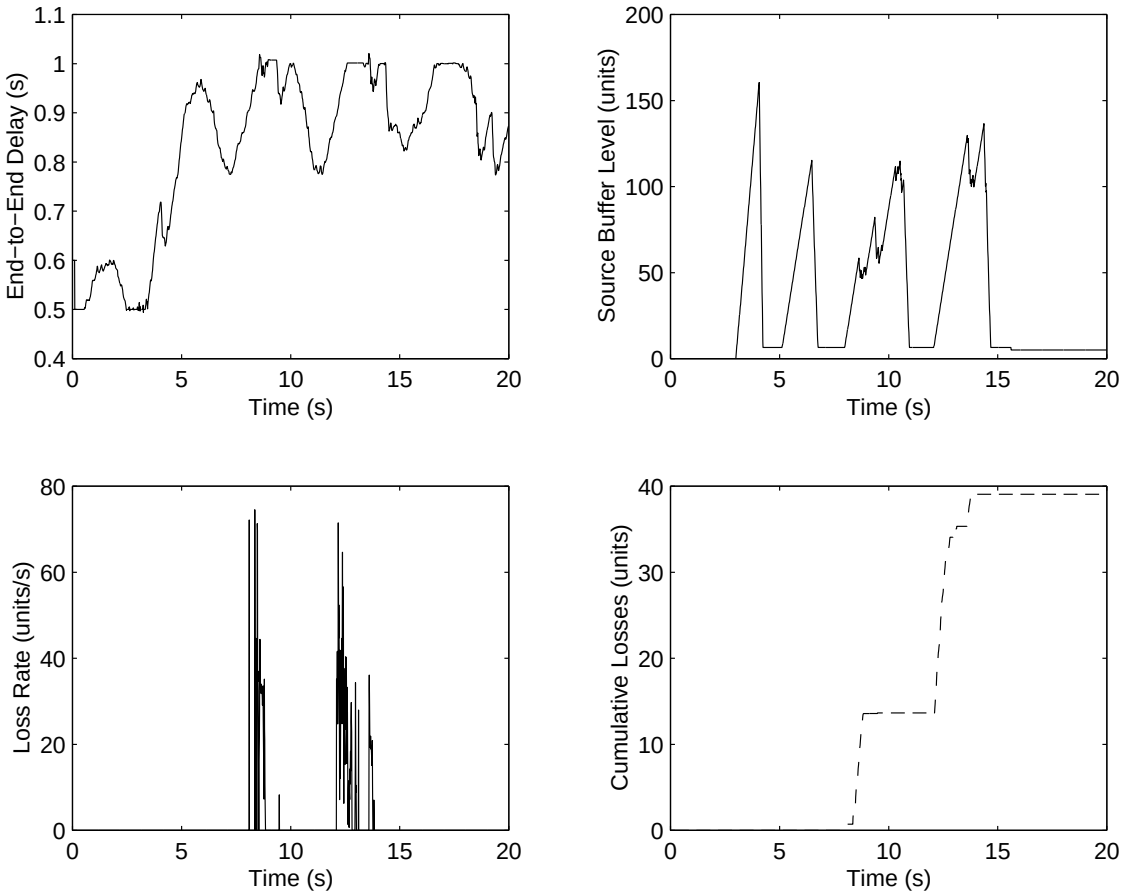

Fig. 27. End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 2 Simulation Using Cross-Traffic 1.

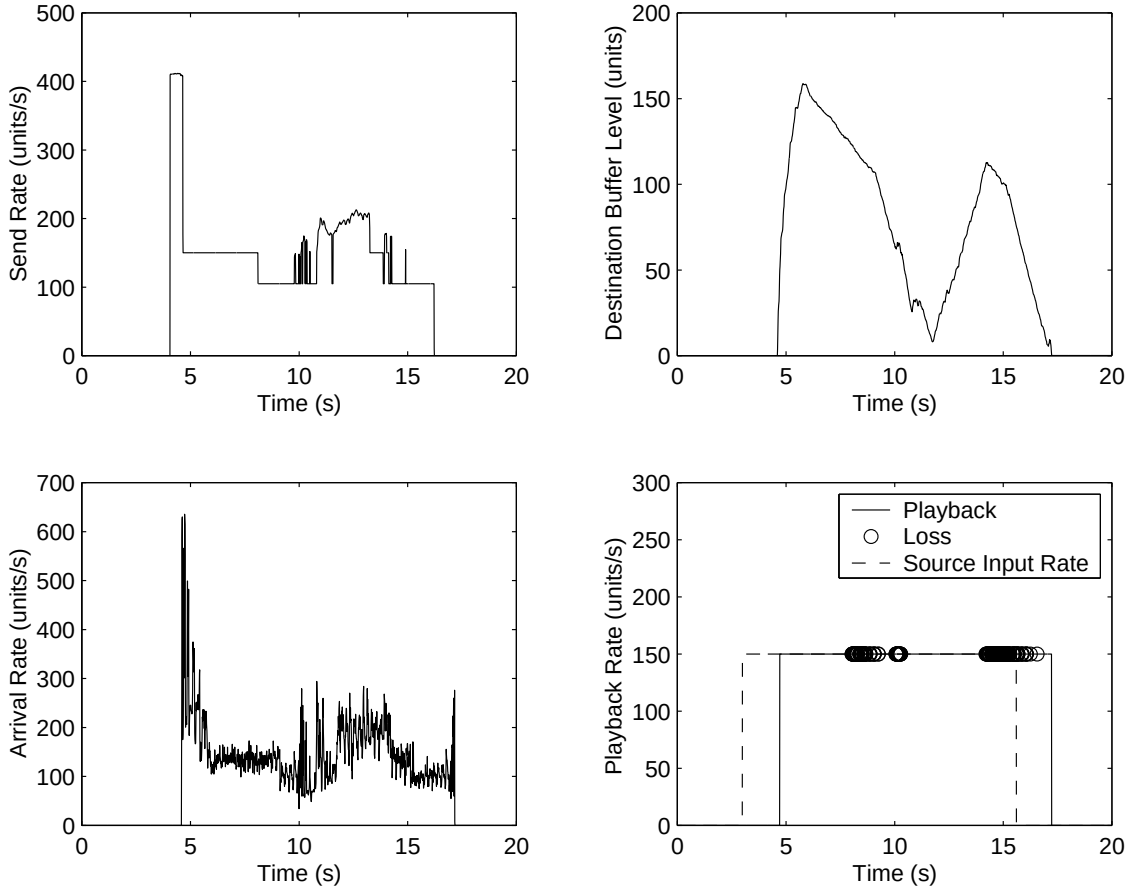

Fig. 28. Buffer Level and Flow Rates for Predictive Controller 2 Simulation Using Cross-Traffic 2.

In the case of the cross-traffic 2 the results are also positive. The losses are decreased by 19% while the dead-time is increased by 17% as shown in Figures 28 and 29.

c. The Effect of the Controller Gain k_2

In Chapter IV, the importance of the controller gain k_2 was discussed. As the gain is decreased the losses are reduced. However, as this gain is decreased playback disruptions are more likely to occur.

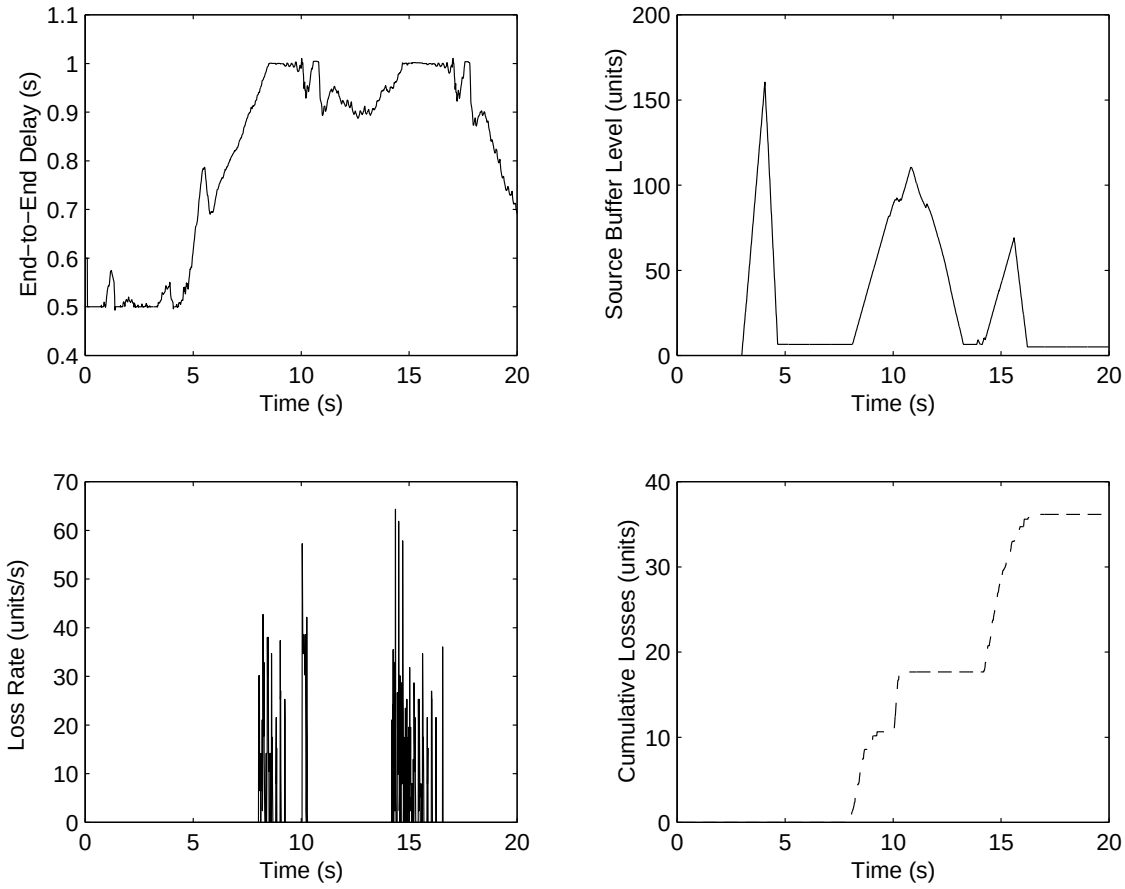

Fig. 29. End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 2 Simulation Using Cross-Traffic 2.

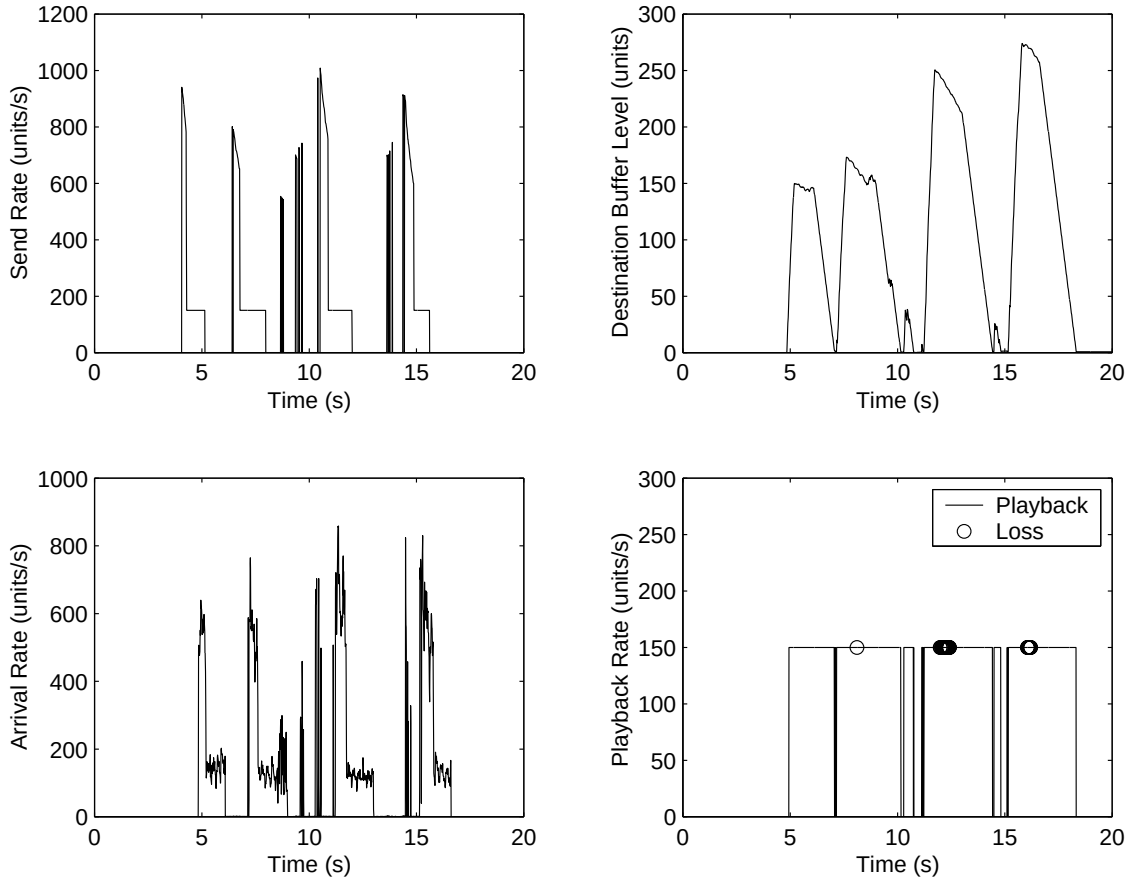

Fig. 30. Buffer Level and Flow Rates for Predictive Controller 2 Simulation Using Cross-Traffic 1 for $k_2 = 0$.

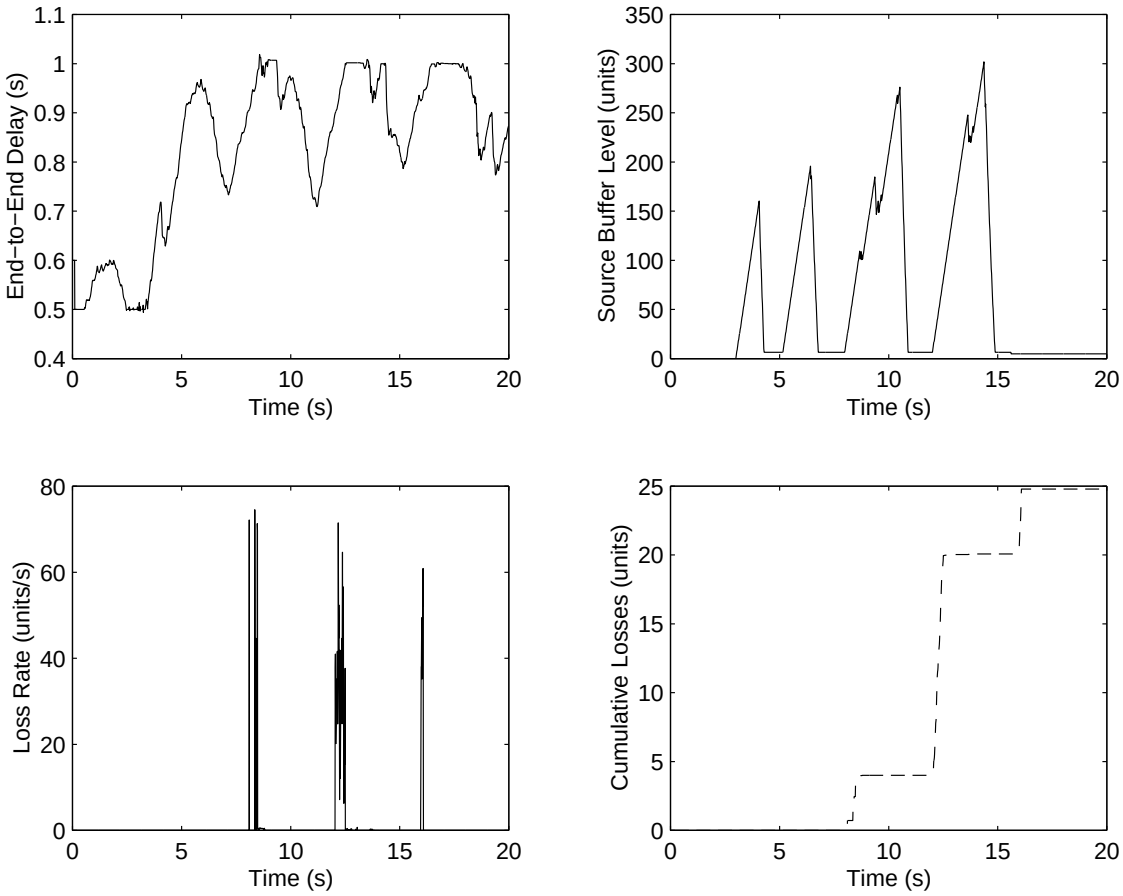

Fig. 31. End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 2 Simulation Using Cross-Traffic 1 for $k_2 = 0$.

Figures 30 and 31 demonstrate this argument. The system response for $k_2 = 0$ is shown. The controller is shut off when high delay is predicted. As one can see, the losses are reduced by 62%. However, the destination buffer empties as it runs out of flow while the controller is off. As a result, playback disruptions occur, and the conclusion is that low values for k_2 should be avoided.

4. Nonlinear Control - Controller 3

Controller 3 uses the accumulation signal as feedback in order to reduce losses. Similarly as in the case of Controller 2, Controller 3 can be reactive or predictive. The accumulation is a signal sensitive to the send rate among other and the delay depends mostly on the cross-traffic since the source send rate is a small portion of the the total flow that goes through the network. Nevertheless, the accumulation depends significantly on the flow level under investigation. This means that the accumulation reflects not only the cross-flow dynamics but also the dynamics of the controlled flow, whereas the delay signal mainly represents the dynamics of the cross-flow. Furthermore, in this series of simulations an additional plot is shown, which depicts the accumulation and the cumulative sending and arrival flow rates.

a. Reactive Implementation of Controller 3

The Figures 32, 33 and 34 show the simulated system response for this controller using cross-traffic 1. The losses are reduced by 16% where as the dead-time is increased by 20% over the uncontrolled case.

In the case of the cross-traffic 2 the controller manages to contribute to a 14% decrease in losses. The dead-time is increased by 20%. The results are demonstrated in Figures 35, 36 and 37

Observe the send rate in Figure 35 and compare it with the accumulation shown

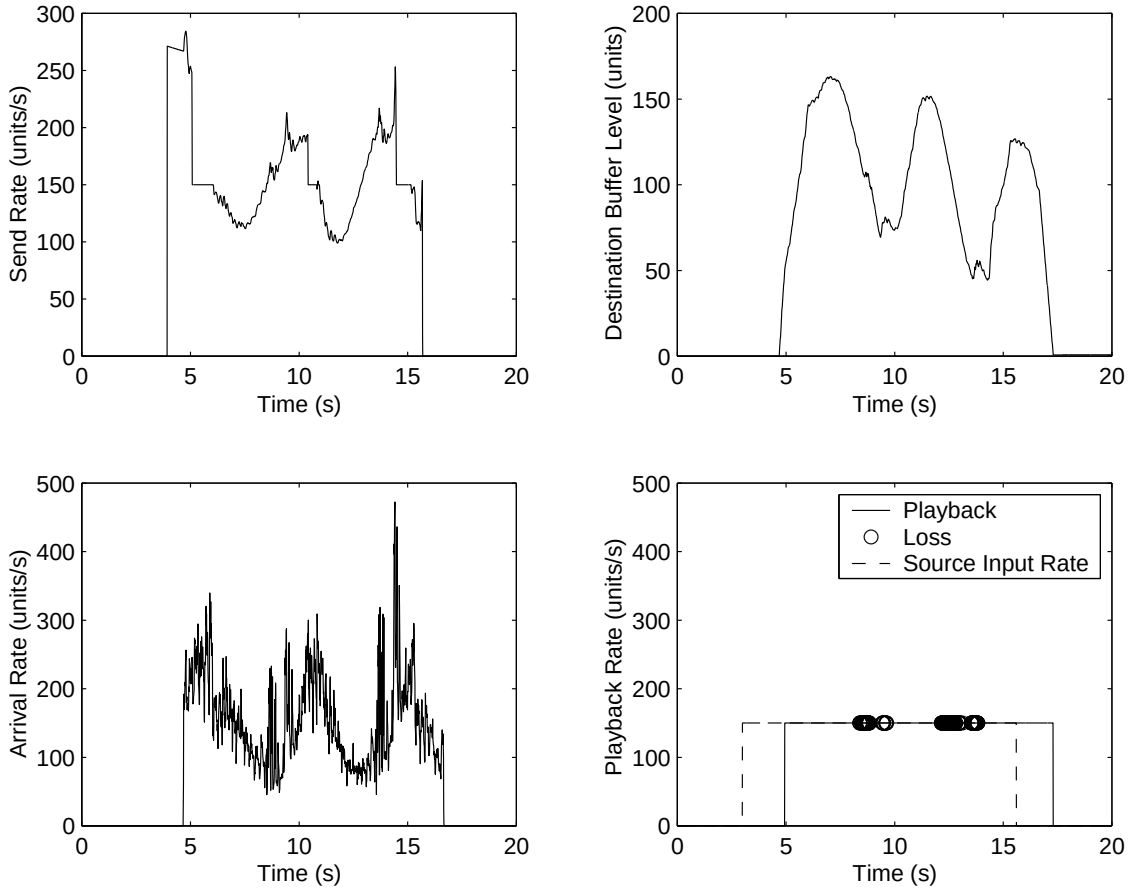

Fig. 32. Buffer Level and Flow Rates for Reactive Controller 3 Simulation Using Cross-Traffic 1.

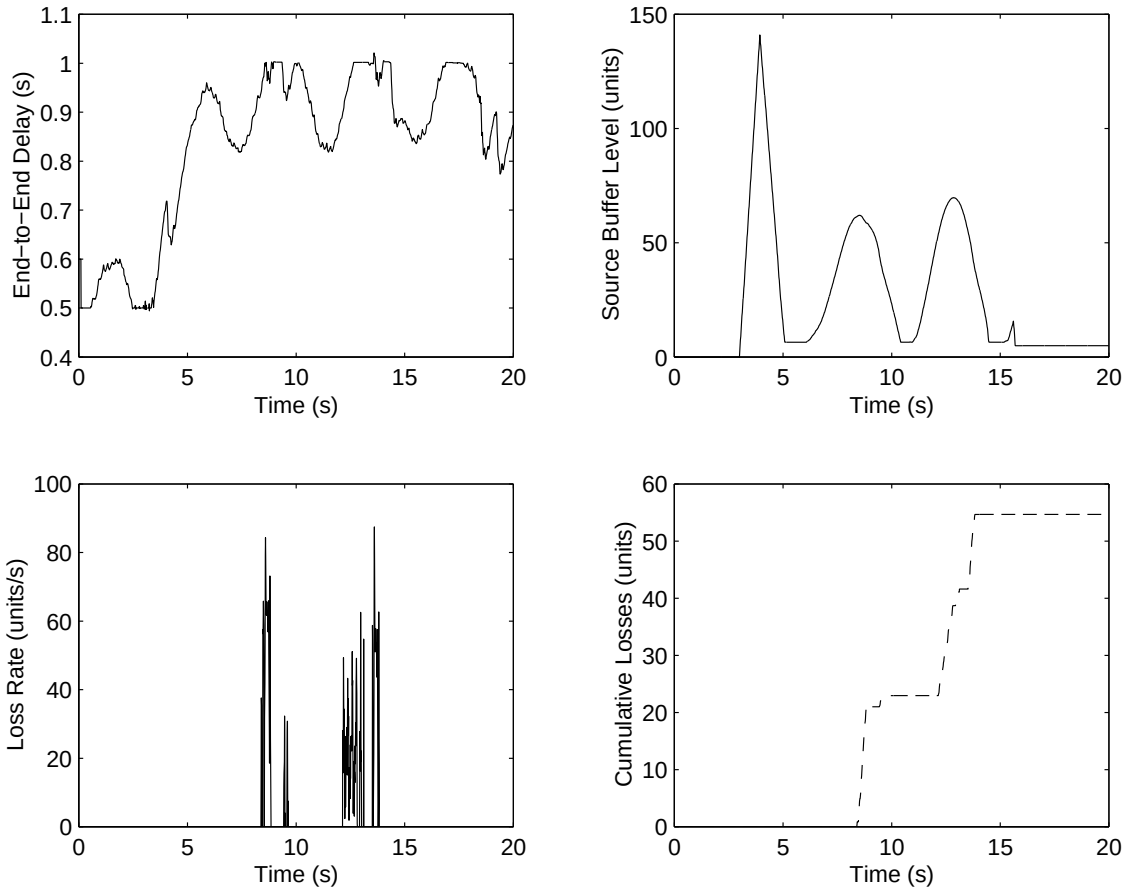

Fig. 33. End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 3 Simulation Using Cross-Traffic 1.

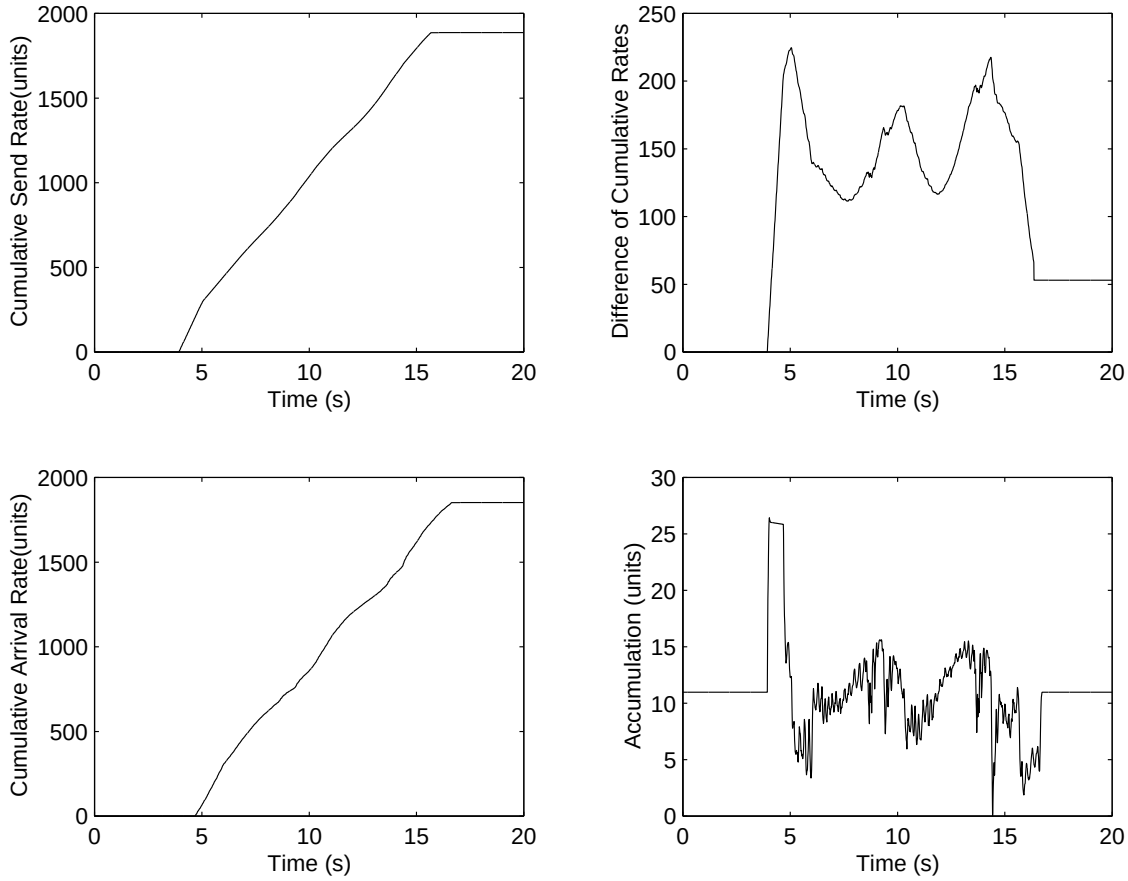

Fig. 34. Cumulative Flow Rates and Accumulation for Reactive Controller 3 Simulation Using Cross-Traffic 1.

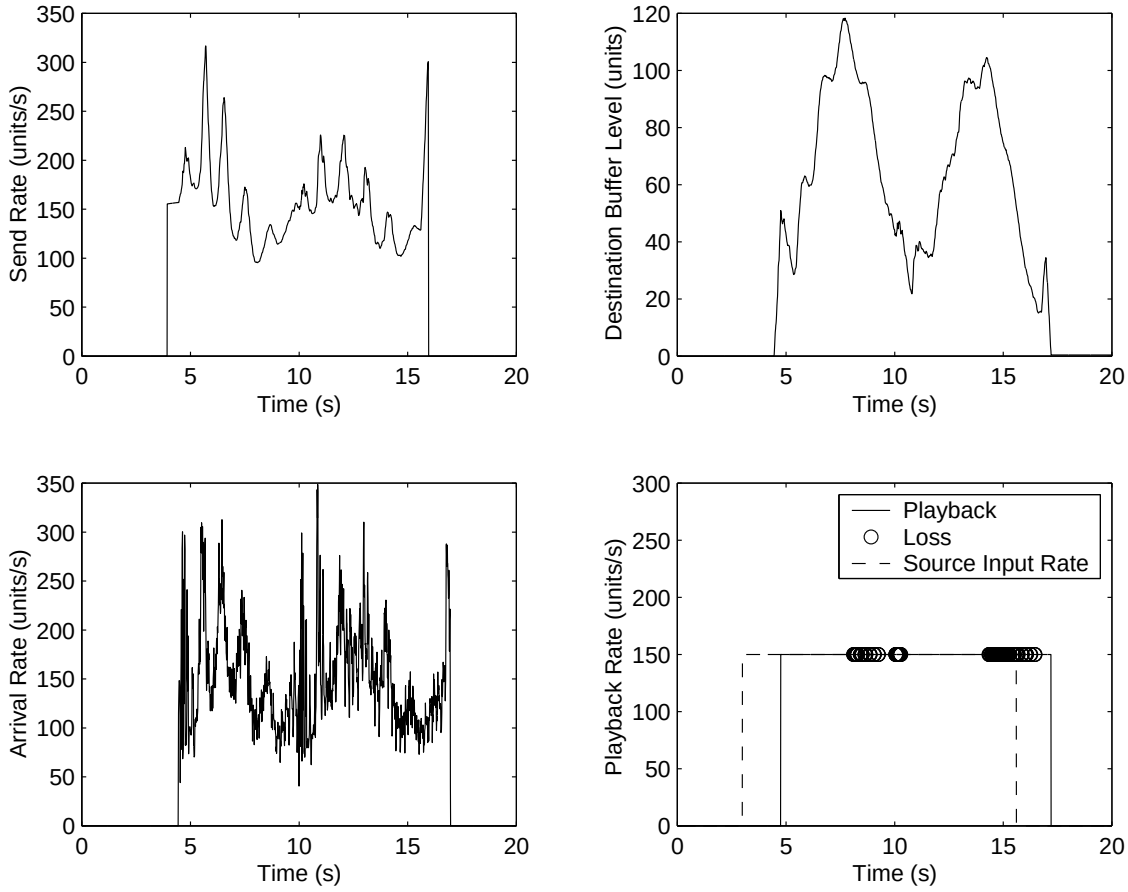

Fig. 35. Buffer Level and Flow Rates for Reactive Controller 3 Simulation Using Cross-Traffic 2.

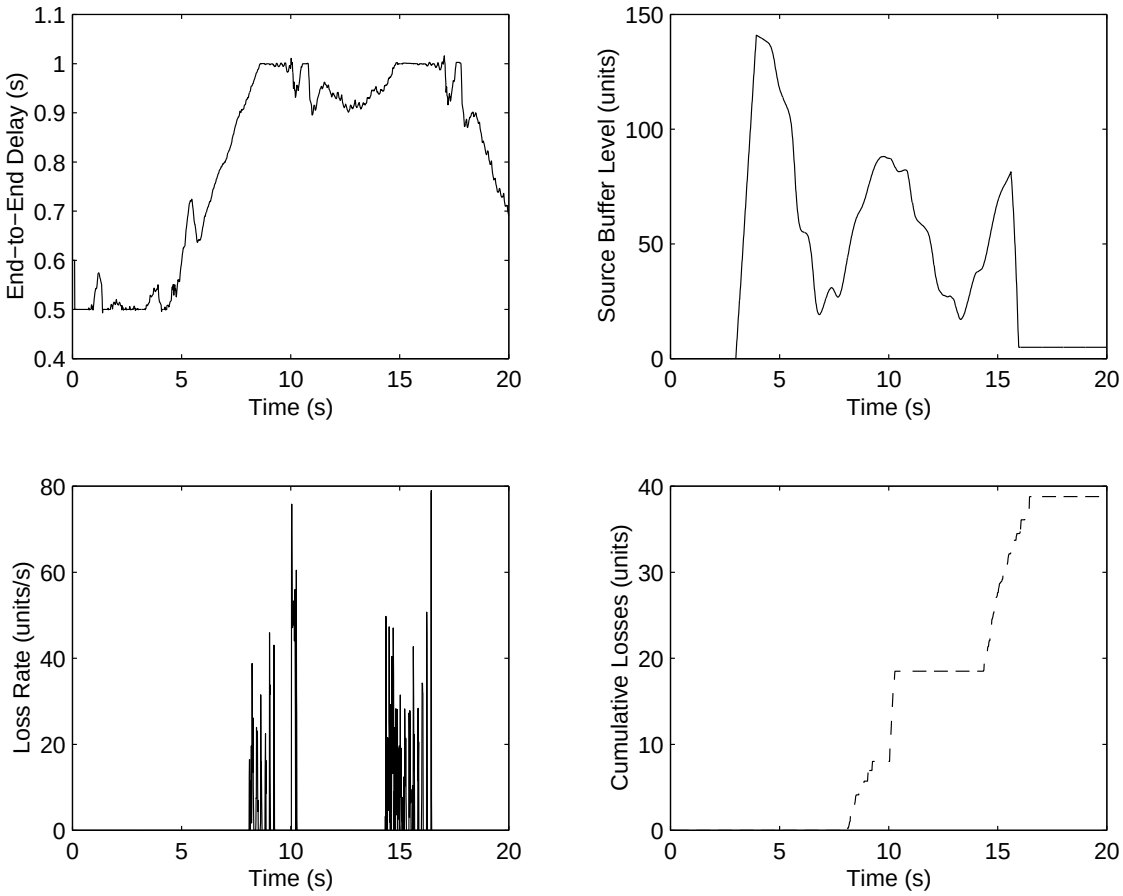

Fig. 36. End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 3 Simulation Using Cross-Traffic 2.

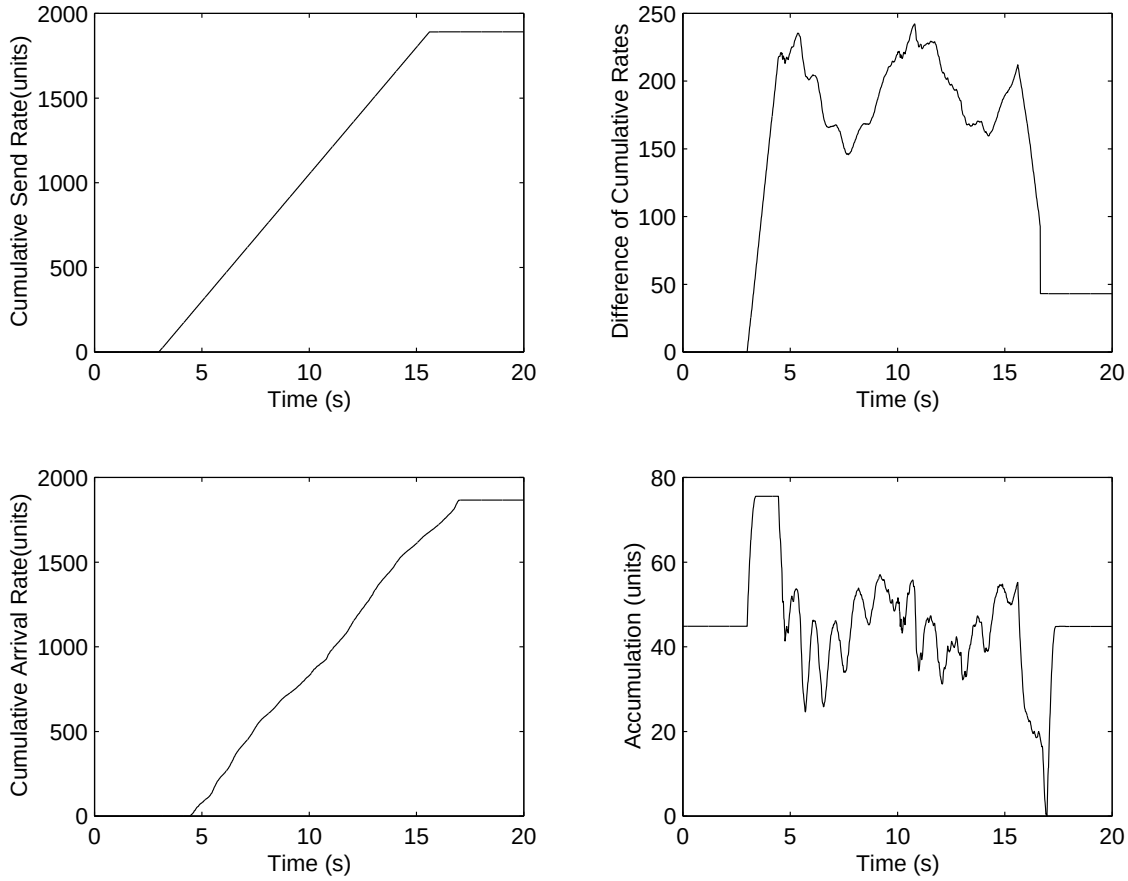

Fig. 37. Cumulative Flow Rates and Accumulation for Reactive Controller 3 Simulation Using Cross-Traffic 2.

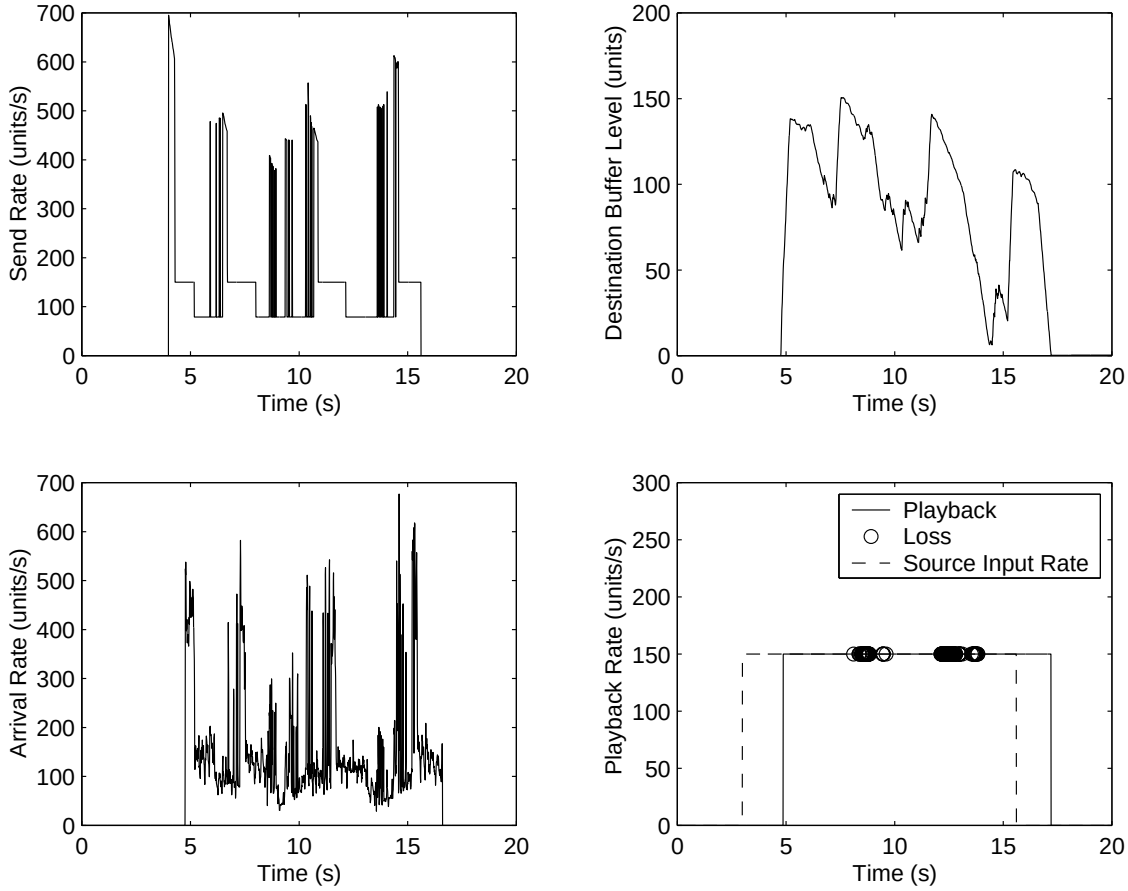

Fig. 38. Buffer Level and Flow Rates for Predictive Controller 3 Simulation Using Cross-Traffic 1.

in Figure 37. The effect of the inverse accumulation term, is clearly shown.

b. Predictive Implementation of Controller 3

Figures 38, 39 and 40 show the system response if future estimates of the delay are used along with cross-traffic 1. The losses are decreased by 33%, while the dead-time is increased by 14%.

The case of cross-traffic 2 is shown in Figures 41, 42 and 43. The losses are decreased by 37% and the dead-time is increased by 32%. This case is similar to the

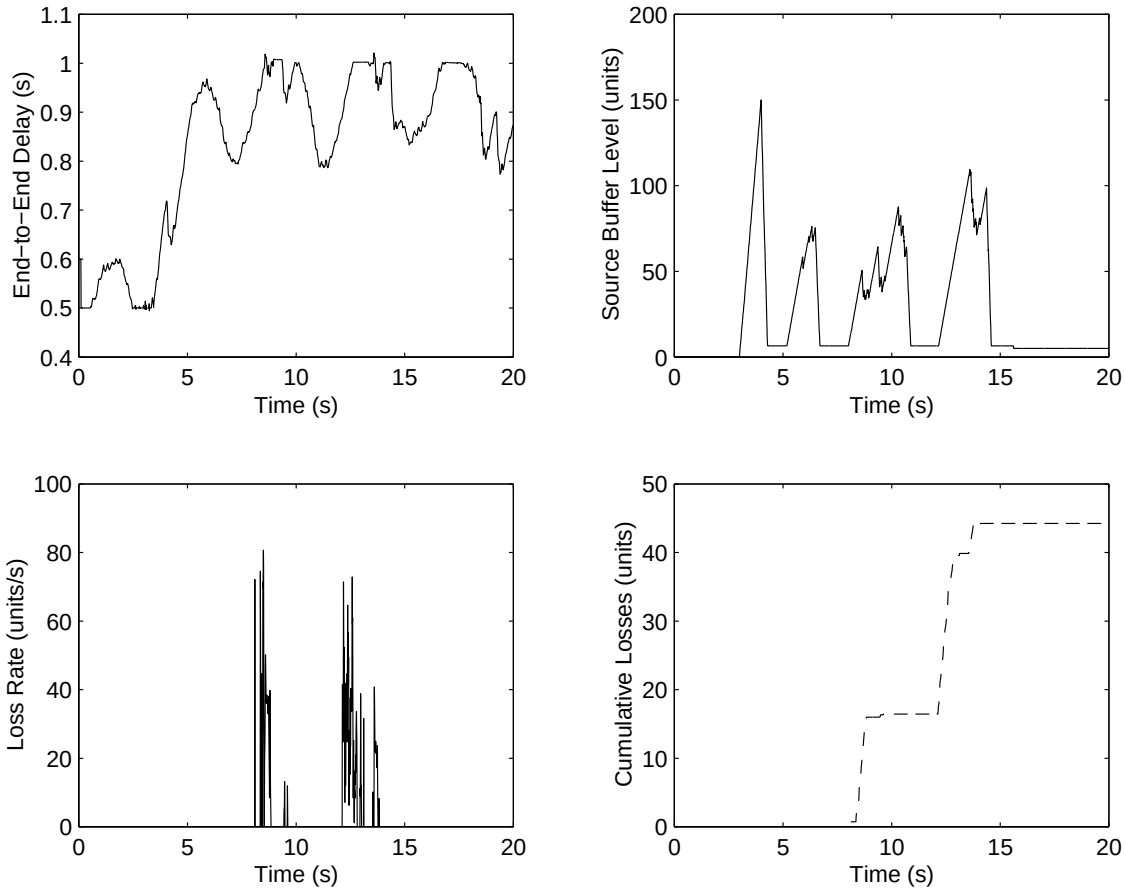

Fig. 39. End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 3 Simulation Using Cross-Traffic 1.

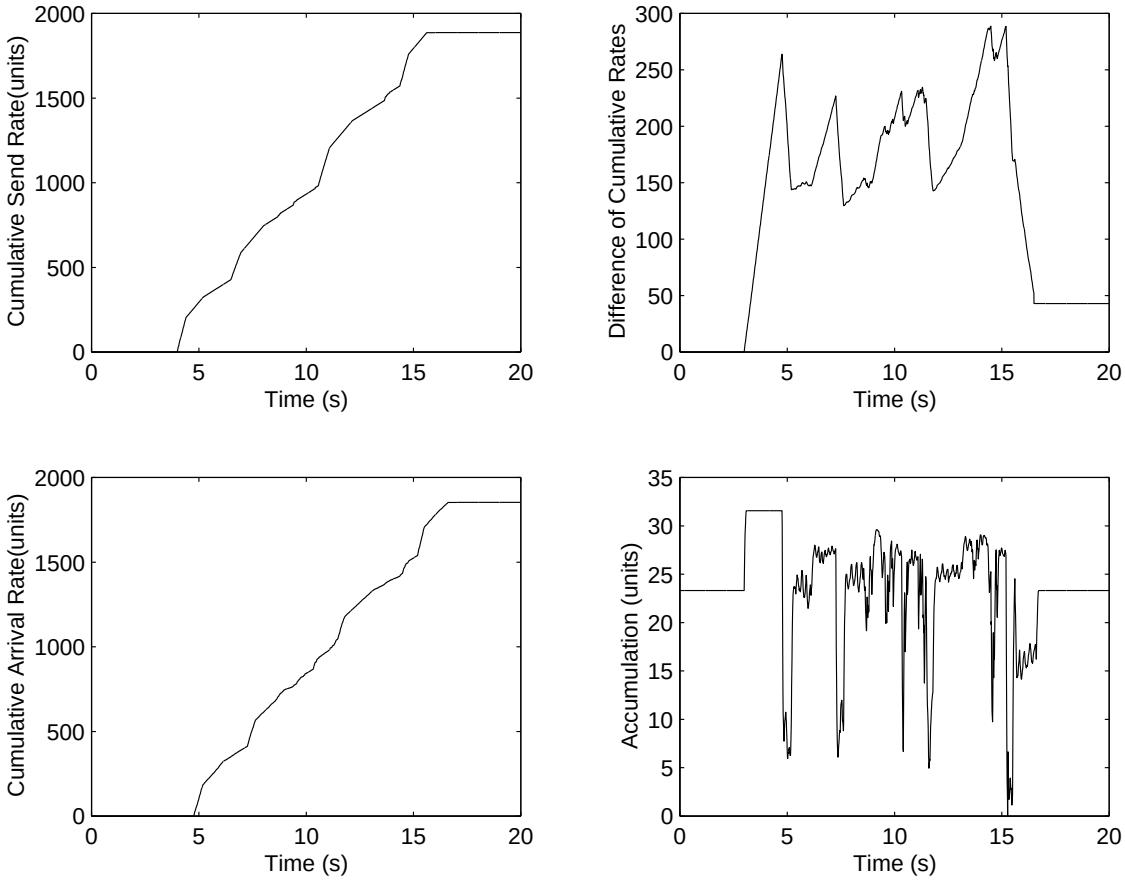

Fig. 40. Cumulative Flow Rates and Accumulation for Predictive Controller 3 Simulation Using Cross-Traffic 1.

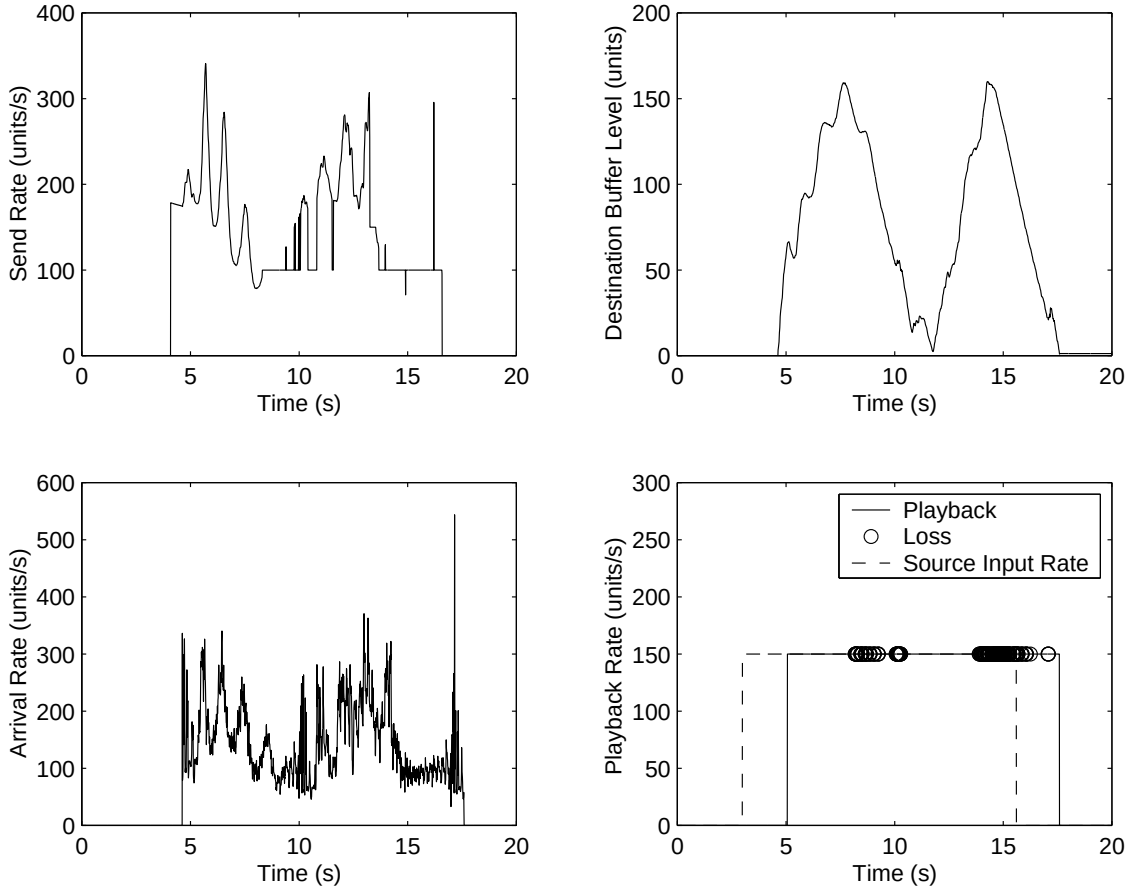

Fig. 41. End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 3 Simulation Using Cross-Traffic 2.

predictively implemented Controller 2. The improvement as compared to the reactive implementation is significant.

5. Model Predictive Control - Controller 4

The MPC algorithm is used, to investigate the impact on the destination buffer level. Simulations with different outputs selected for control are also presented. At first, the empirical model of the open-loop system is identified for a specific cross-traffic trace. Once the input output relationship is known, the implementation of the algorithm

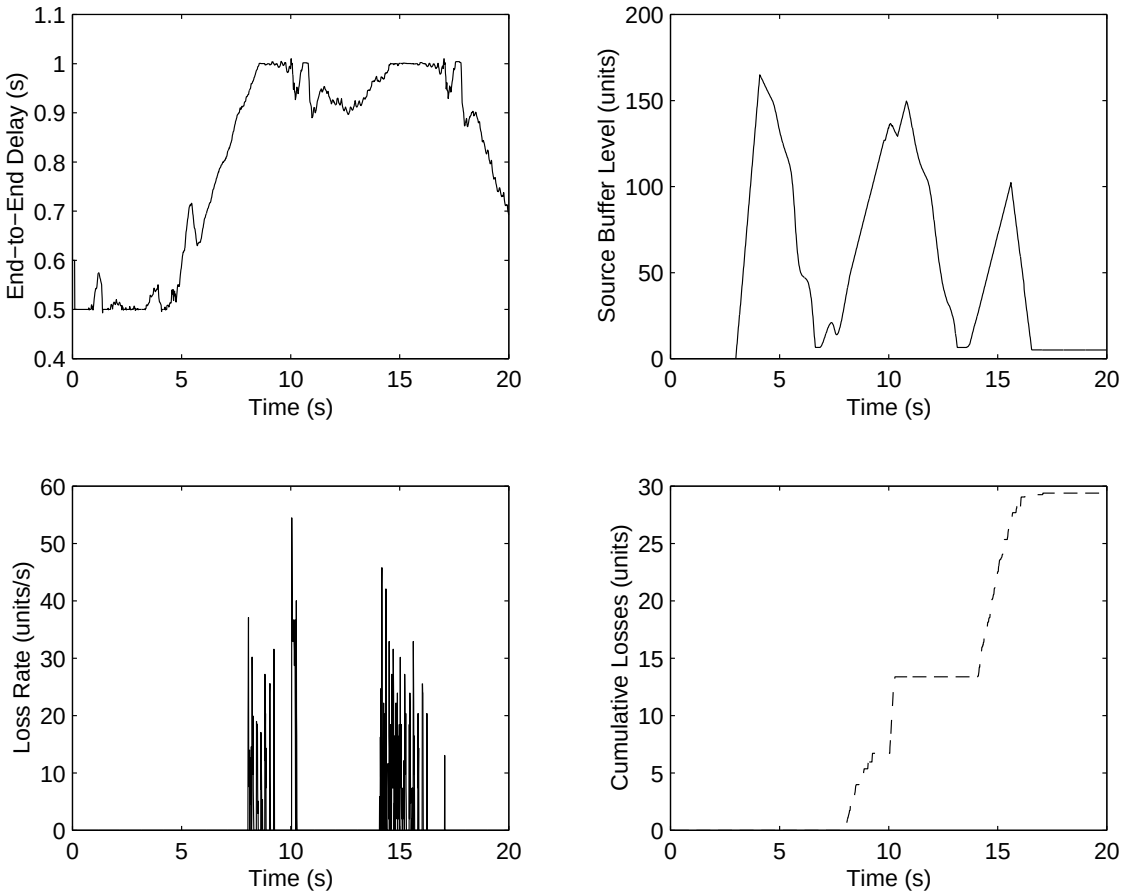

Fig. 42. End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 3 Simulation Using Cross-Traffic 2.

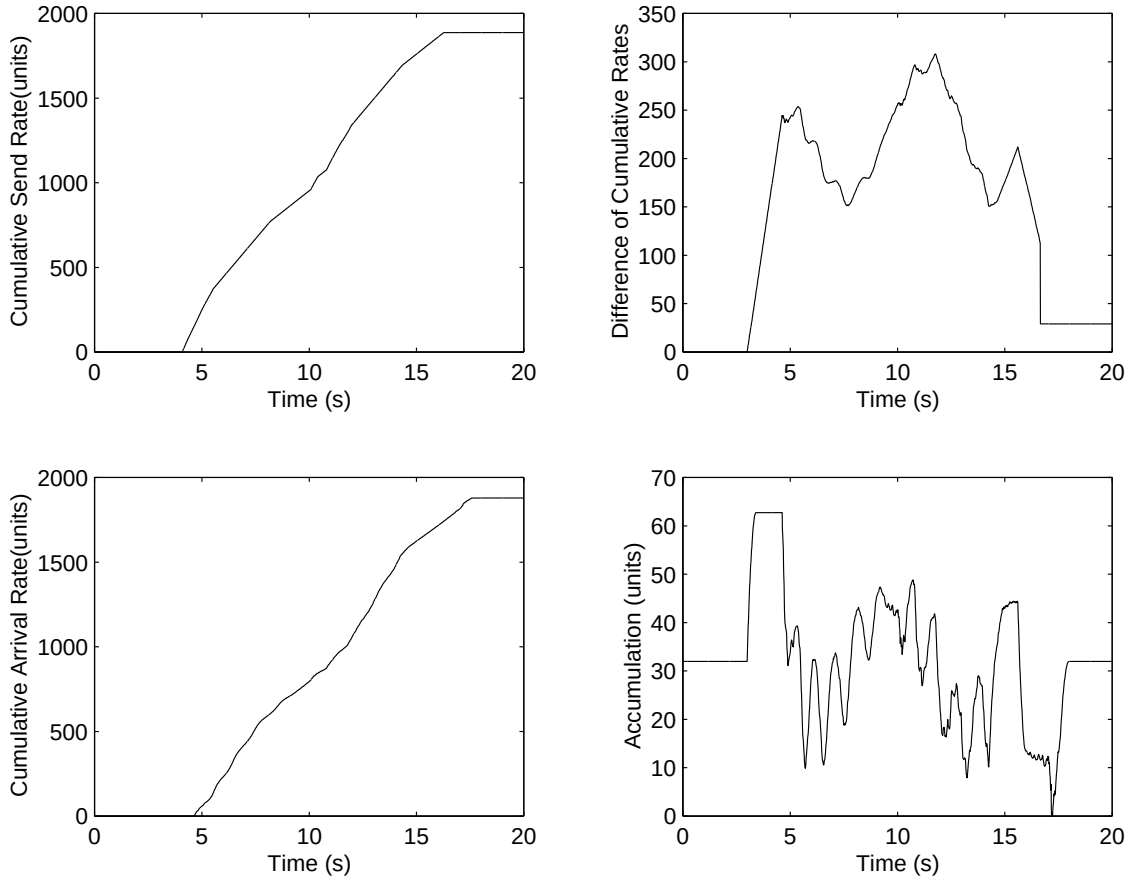

Fig. 43. Cumulative Flow Rates and Accumulation for Predictive Controller 3 Simulation Using Cross-Traffic 2.

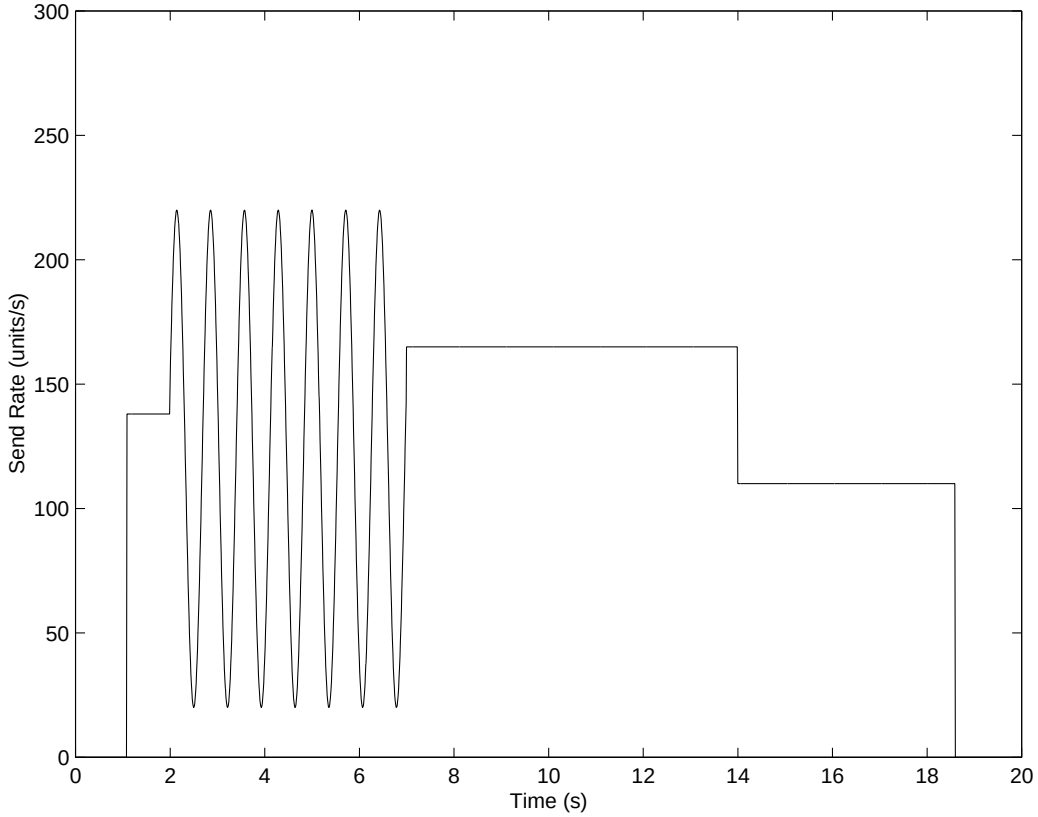

Fig. 44. Send Rate Used to Identify the Open-Loop System Model.

represented by Equation (4.18) is straightforward. The receding prediction horizon is selected to be 2 seconds. The sampling period, is selected to be 100 ms. This means that the furthest ahead prediction will be 20 step-ahead. The function shown in Figure 44, is applied initially as the system input to identify the open-loop system.

Defining the destination buffer as the system output, the prediction is tested for a constant send rate which is data new to the predictor. The Figures 45 and 46 show the prediction for both cross-traffic traces.

Two types of errors are used as performance metrics of the developed predictors. The first is called *Maximum Relative Error* (MRE) and is defined as follows:

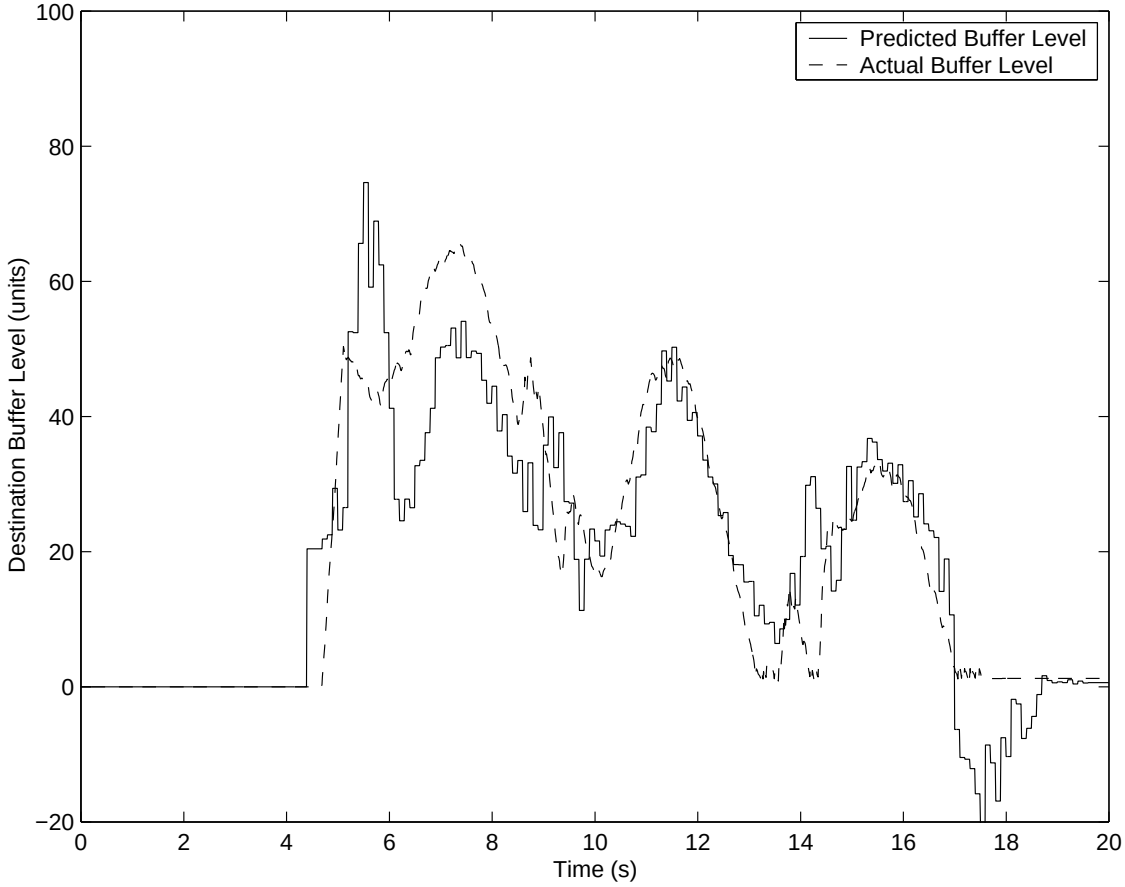

Fig. 45. 20-Step-Ahead Prediction of Destination Buffer Level Using Cross-Traffic 1 for Application Send Rate of 150 ups.

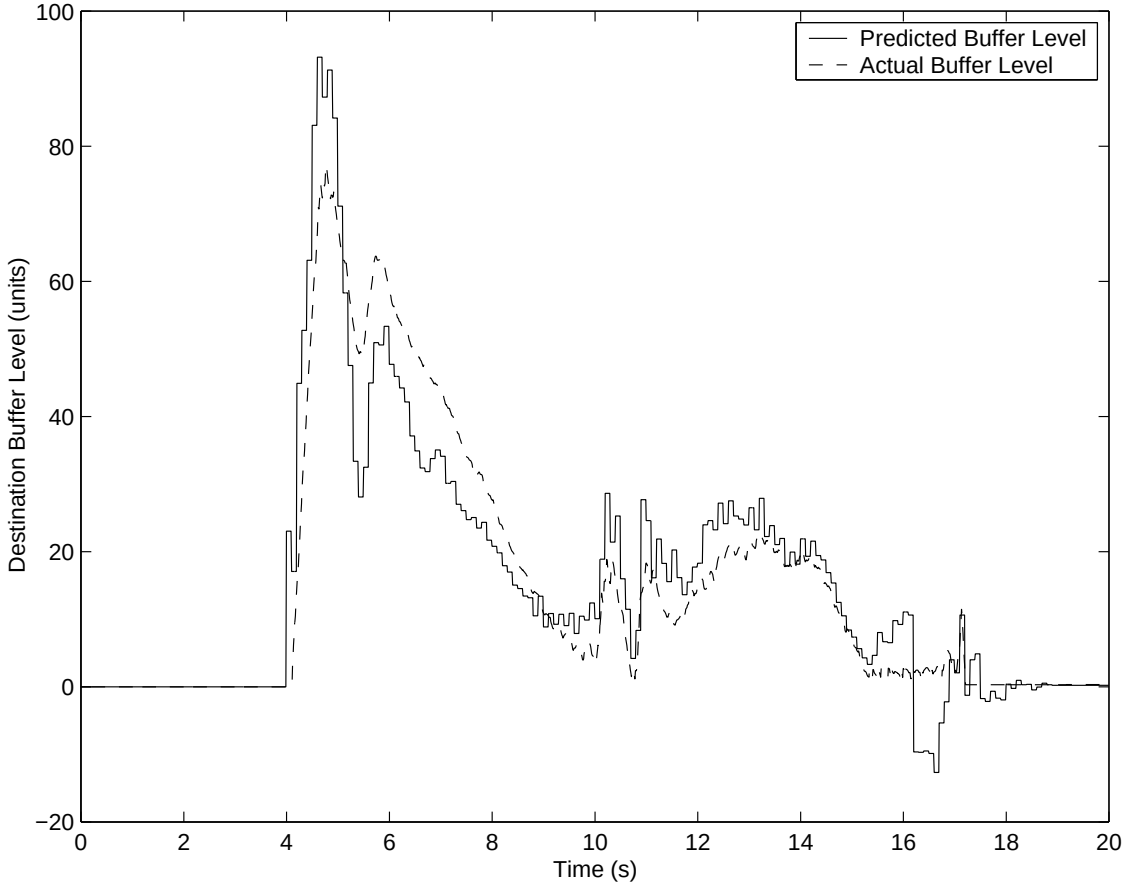

Fig. 46. 20-Step-Ahead Prediction of Destination Buffer Level Using Cross-Traffic 2 for Application Send Rate of 150 ups.

Table II. 20-Step-Ahead Prediction Errors

Send Rate (ups)	**Cross-Traffic**	**MRE (%)**	**ARE (%)**
150	Cross-Traffic 1	53	10.1
150	Cross-Traffic 2	75	13.3
120	Cross-Traffic 1	48	11.3
180	Cross-Traffic 1	49	10.4

$$MRE = \max_{1 \leq k \leq N} |\frac{y(k) - \hat{y}(k + p/k)}{y(k)}| \times 100, \tag{5.1}$$

where N is the total number of data points, $y(k)$ the output observation, $\hat{y}(k + p/k)$ the predicted output and p is the prediction horizon. It is calculated after ignoring the first set-up period. During this period, the predictor does not perform well, until a full set of input-output measurements is obtained. This error is useful for identifying the regions where the predictor fails.

The other error is called *Average Relative Error* (ARE) and gives a better indication for the prediction. It is defined by

$$ARE = \frac{1}{N} \sum_{k=1}^{N} |\frac{y(k) - \hat{y}(k + p/k)}{y(k)}| \times 100. \tag{5.2}$$

The sensitivity of the prediction on send rate is also tested. In this work, the predictor is tested for different constant rates. Figures 47 and 48 show the prediction for send rates of 120 ups and 180 ups respectively.

Table II summarizes the results. It shows that the error does not change drastically for different data and for different cross-traffic traces.

The prediction is based on Equation (4.18). For a given future input the predicted

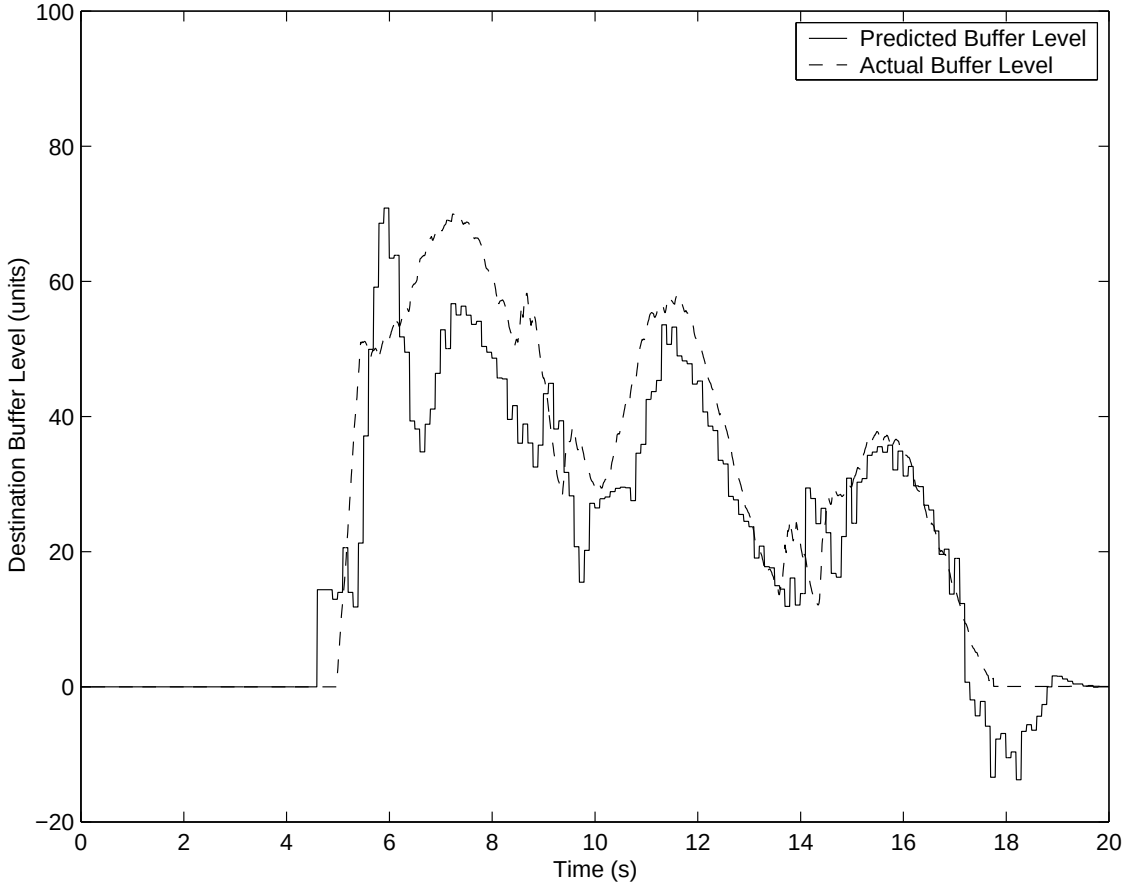

Fig. 47. 20-Step-Ahead Prediction of Destination Buffer Level Using Cross-Traffic 1 for Application Send Rate of 120 ups.

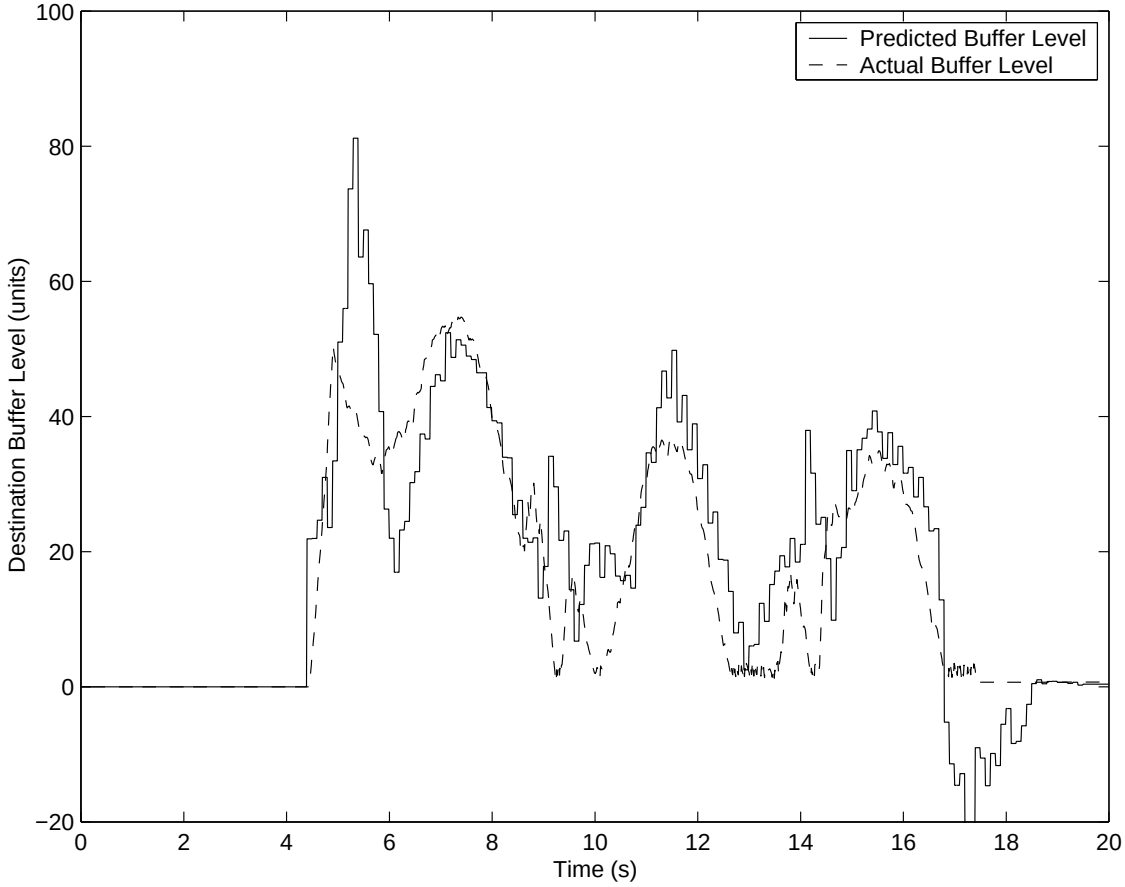

Fig. 48. 20-Step-Ahead Prediction of Destination Buffer Level Using Cross-Traffic 1 for Application Send Rate of 180 ups.

output is calculated. The prediction is updated at each simulation step. Therefore, at each simulation step Equation (4.18) calculates a vector of the next 20 future outputs. This is the furthest ahead prediction. At the next simulation step the procedure is repeated and again the 19 intermediated predictions are skipped and only the last prediction is plotted. Figures 45, 46, 47 and 48 show the 20th-step ahead (which is the worst) prediction for different input data and different cross-traffic traces.

The MPC control method is implemented with an initial open-loop set up phase. This short period during the first few seconds of the media transport allows for buffer level measurements to be received by the source and used in the first few steps of prediction. Once an initial minimum buffer level, x_{on}, is reached, then playback begins and the regular control algorithm is used to determine the send rate.

Figures 49 and 50 demonstrate the results of this simulation which intends to regulate the destination buffer level at a constant desired value. There is a 0% change on the losses and a 18% increase in dead-time. For cross-traffic 2 the Figures 51 and 52 show the system response. There is a 0% change on the losses and a 10% increase in dead-time over the open-loop case.

Observe how well the MPC controller regulates the destination buffer at a constant desired level. There is no impact on losses as regulating the destination buffer level only intends to eliminate disruptions. Nevertheless, the impact of the controller on the output is evident.

One should keep in mind that the MPC simulations show how the specific control algorithm can affect the system output, whatever this is. However, in order for a controller to be considered beneficial a reduction in the losses should be achieved. This could be attempted by defining as an output a signal which includes loss information.

A signal that incudes loss information is the cumulative difference between sending and arrival rate, $\lambda(t)$. The signal has an increasing trend due to the cumulative

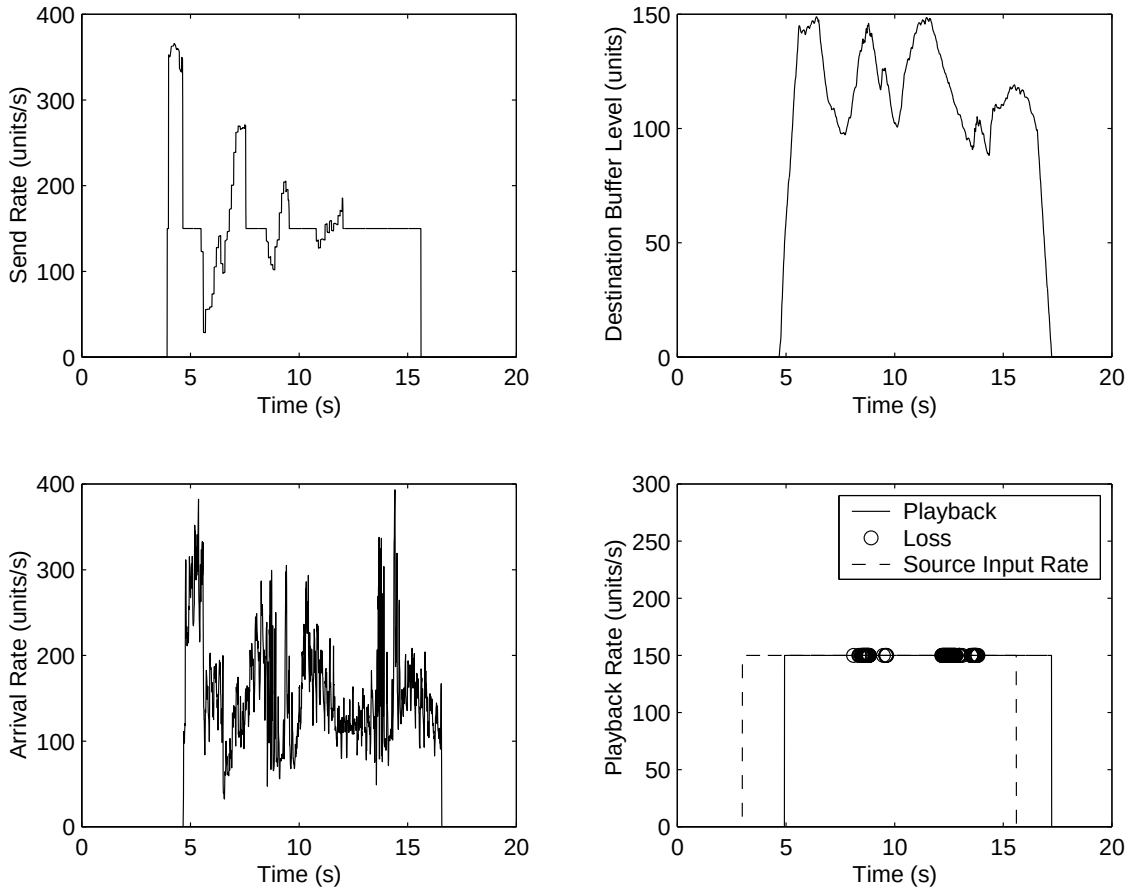

Fig. 49. Buffer Level and Flow Rates for MPC Simulation Using Cross-Traffic 1; Destination Buffer Regulation.

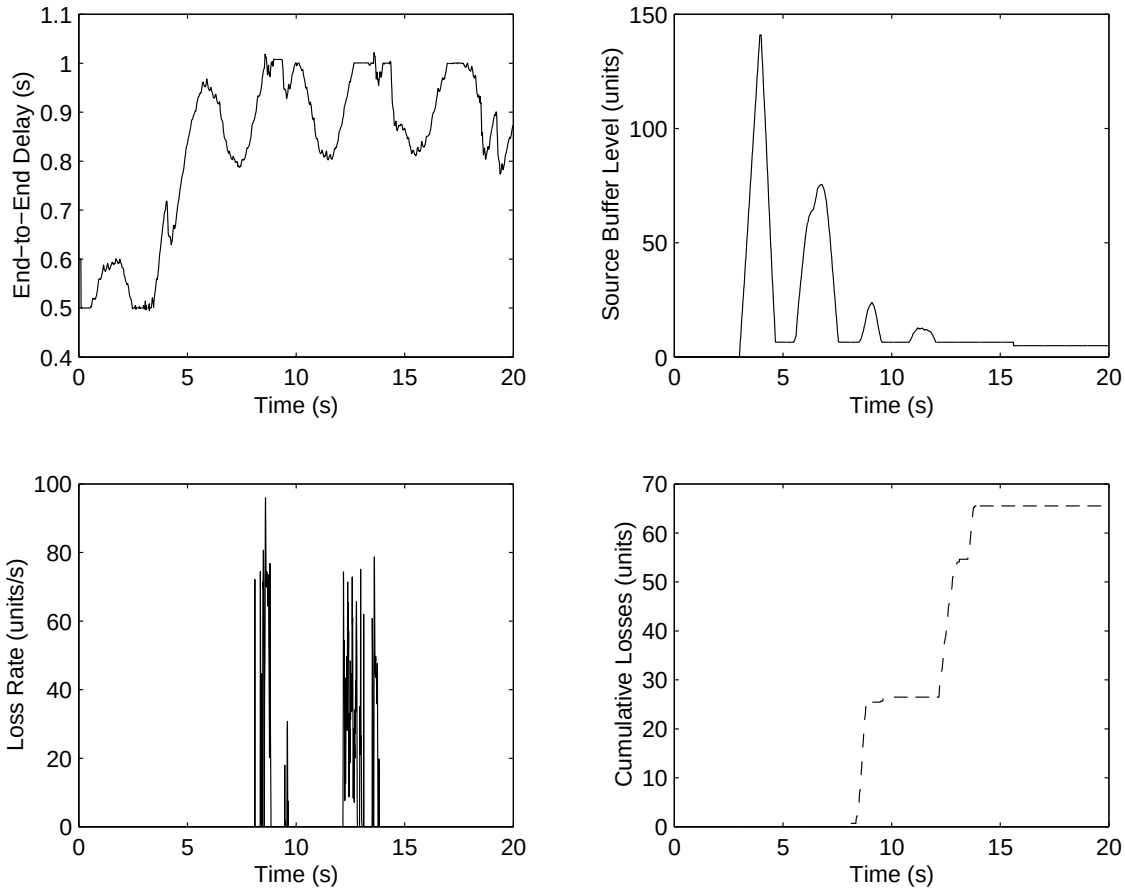

Fig. 50. End-to-End Delay, Source Buffer Level and Losses for MPC Simulation Using Cross-Traffic 1; Destination Buffer Regulation.

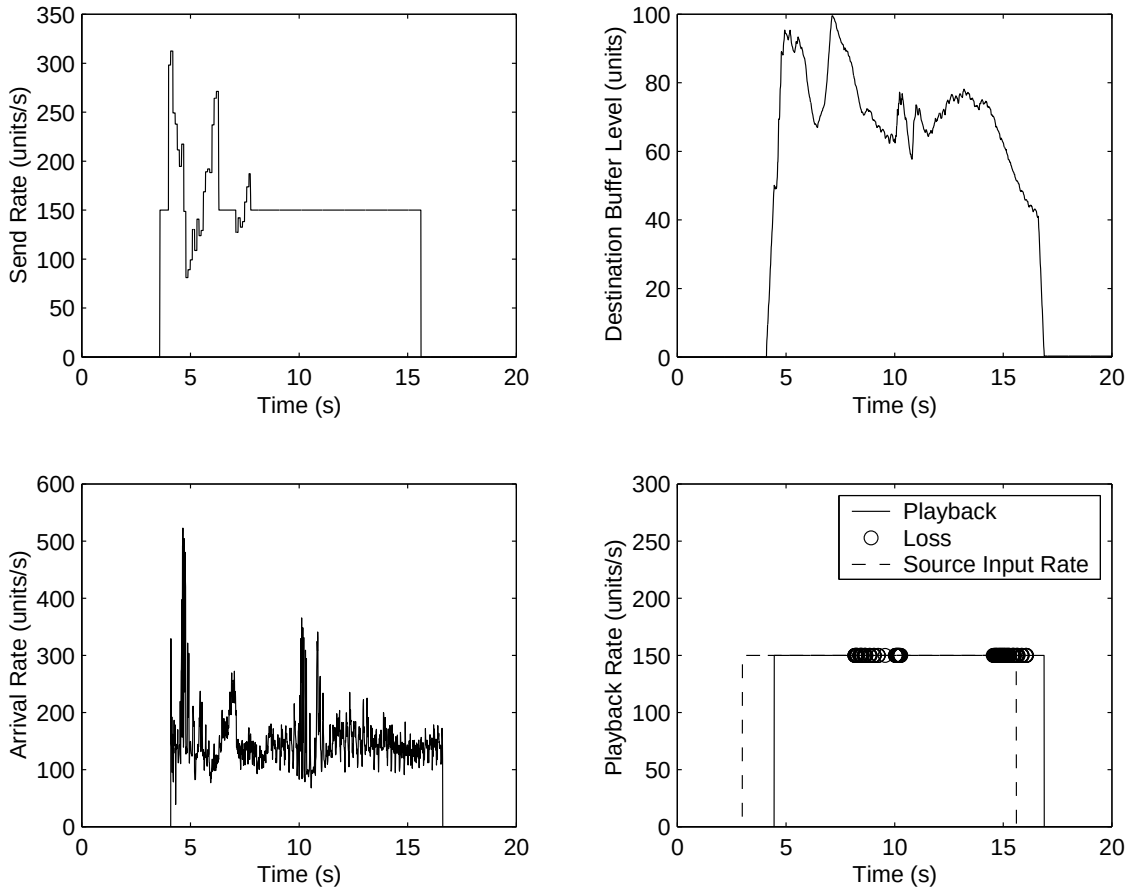

Fig. 51. Buffer Level and Flow Rates for MPC Simulation Using Cross-Traffic 2; Destination Buffer Regulation.

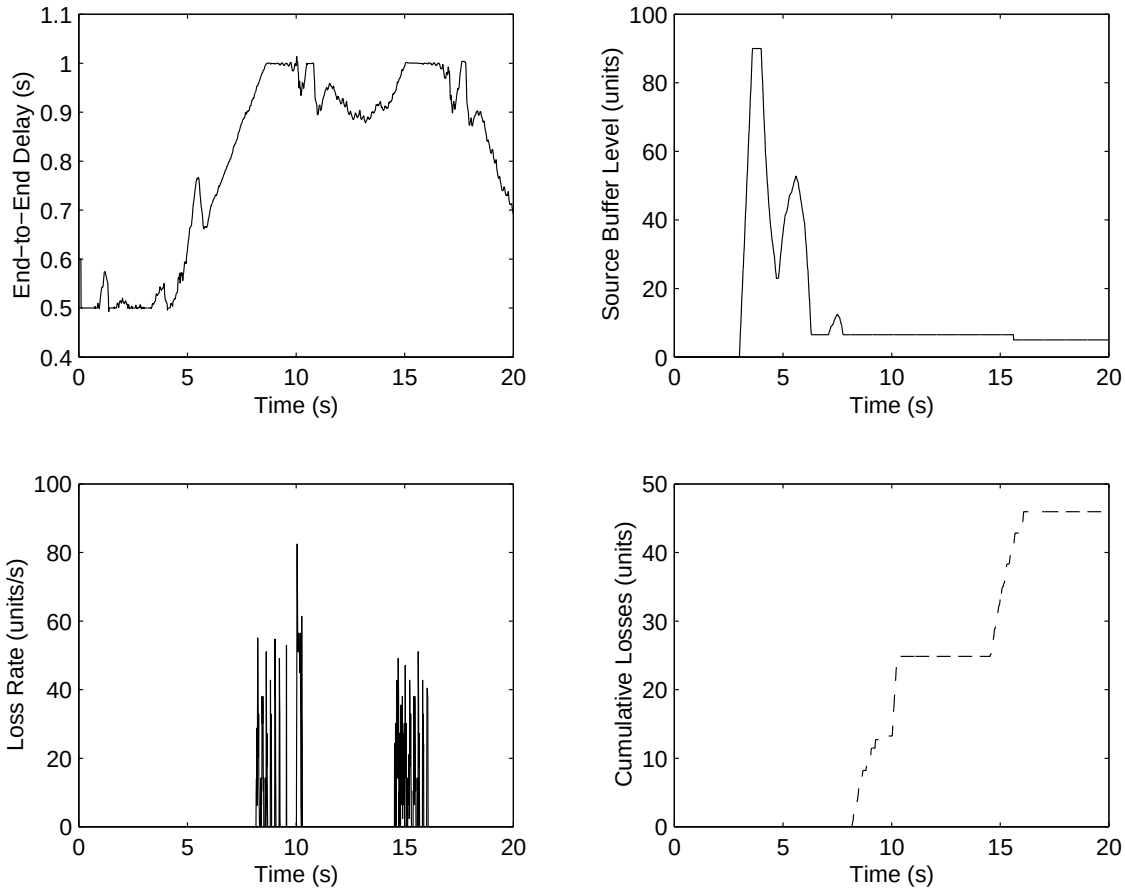

Fig. 52. End-to-End Delay, Source Buffer Level and Losses for MPC Simulation Using Cross-Traffic 2; Destination Buffer Regulation.

losses which are included in $\lambda(t)$. Then instead of the destination buffer, $\lambda(t)$ is used as the system output. The objective is to make the specific signal flat and therefore reduce losses.

Figures 53, 54 and 55 show the system response using $\lambda(t)$ as the controlled signal. Although the controller achieves its objective the losses are not affected. The change was 0% compared to the open-loop case. If Figure 55 is compared to Figure 14 one can see that the cumulative difference when the MPC controller is implemented is almost flat. Keeping $\lambda(t)$ flat was the controller's objective and was successfully reached with a small increase in dead-time of 0.14 (9%) seconds. However, the signal used as an output includes information about the integral of the losses and does not warn the controller when they will occur. Furthermore, the system output $\lambda(t)$ is a combination of two signals as it represents the sum of the accumulation and the losses. Trying to regulate a signal including both losses and accumulation seems infeasible. Separating the two signals and regulating each one with different predictors should be a future step.

This case opens some interesting questions and shows that one system output is not adequate, for the MPC controller to achieve all the objectives of the control problem. The question of course, is how many outputs should be regulated and which ones.

The predicted and actual output signals are shown in Figure 56. As one can see the predictor works well and the predicted output is close to the actual. Reconsider that the prediction shown, is the 20-step ahead prediction. The controller uses all the intermediate predictions as shown in Equation (4.18).

The MPC controller is effective and manages to achieve its goal. However, the objectives of the problem presented in this thesis are more than one, and multiple terms should be used in one control equation, each one contributing to a different

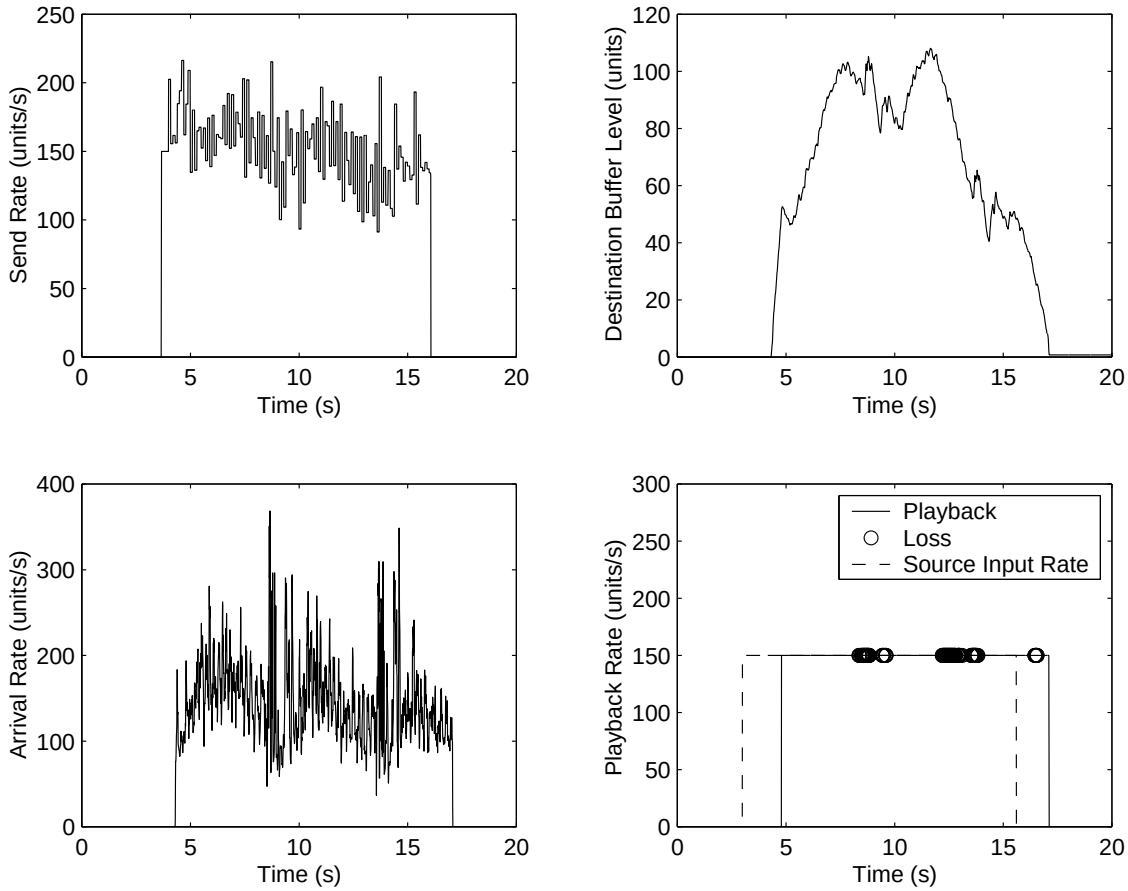

Fig. 53. Buffer Level and Flow Rates for MPC Simulation Using Cross-Traffic 1; Cumulative Flow Difference Regulation.

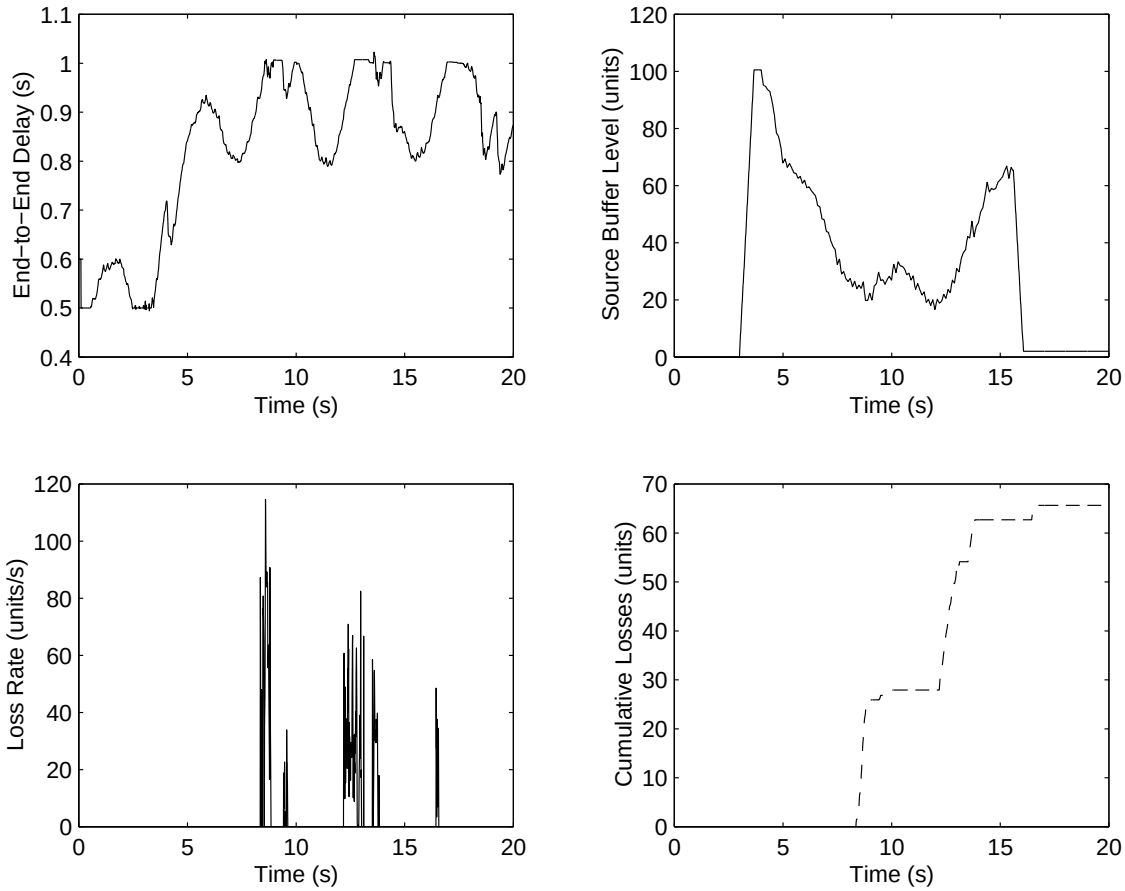

Fig. 54. End-to-End Delay, Source Buffer Level and Losses for MPC Simulation Using Cross-Traffic 1; Cumulative Flow Difference Regulation.

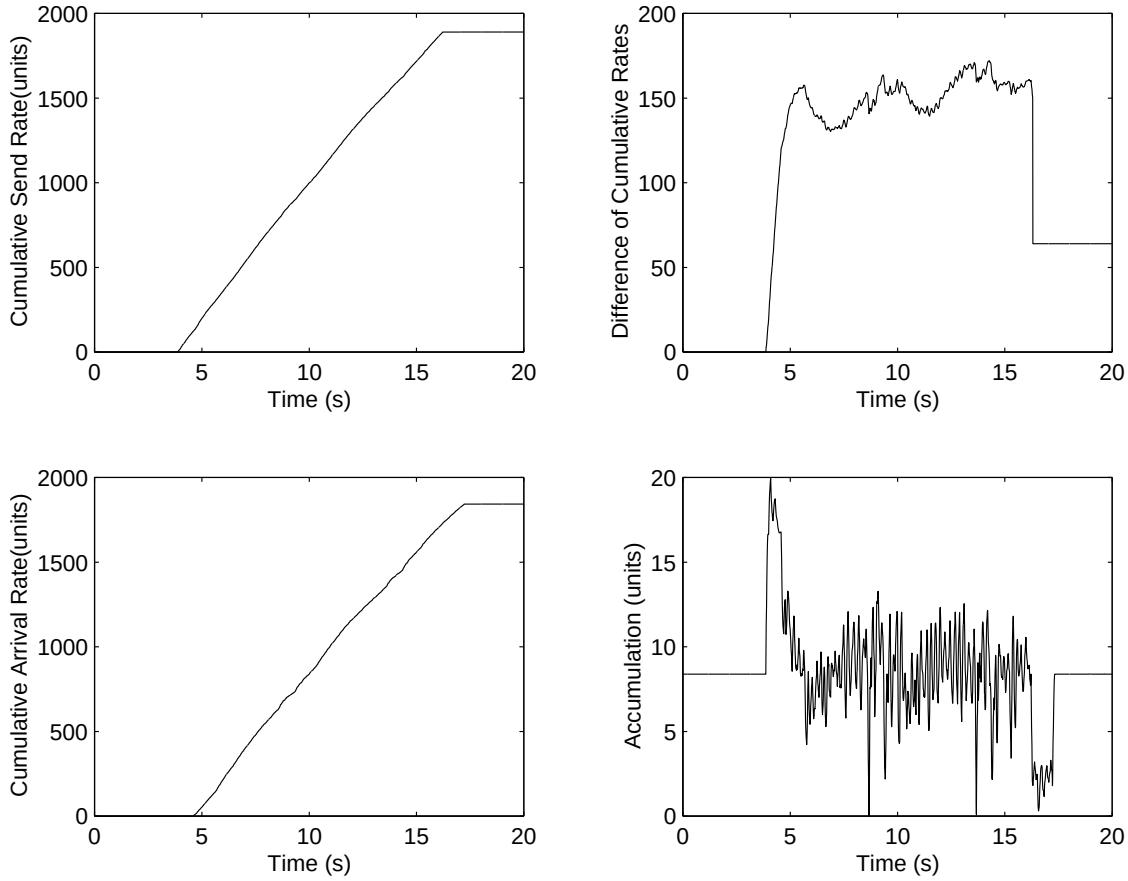

Fig. 55. Cumulative Flow Rates and Accumulation for MPC Simulation Using Cross-Traffic 1; Cumulative Flow Difference Regulation.

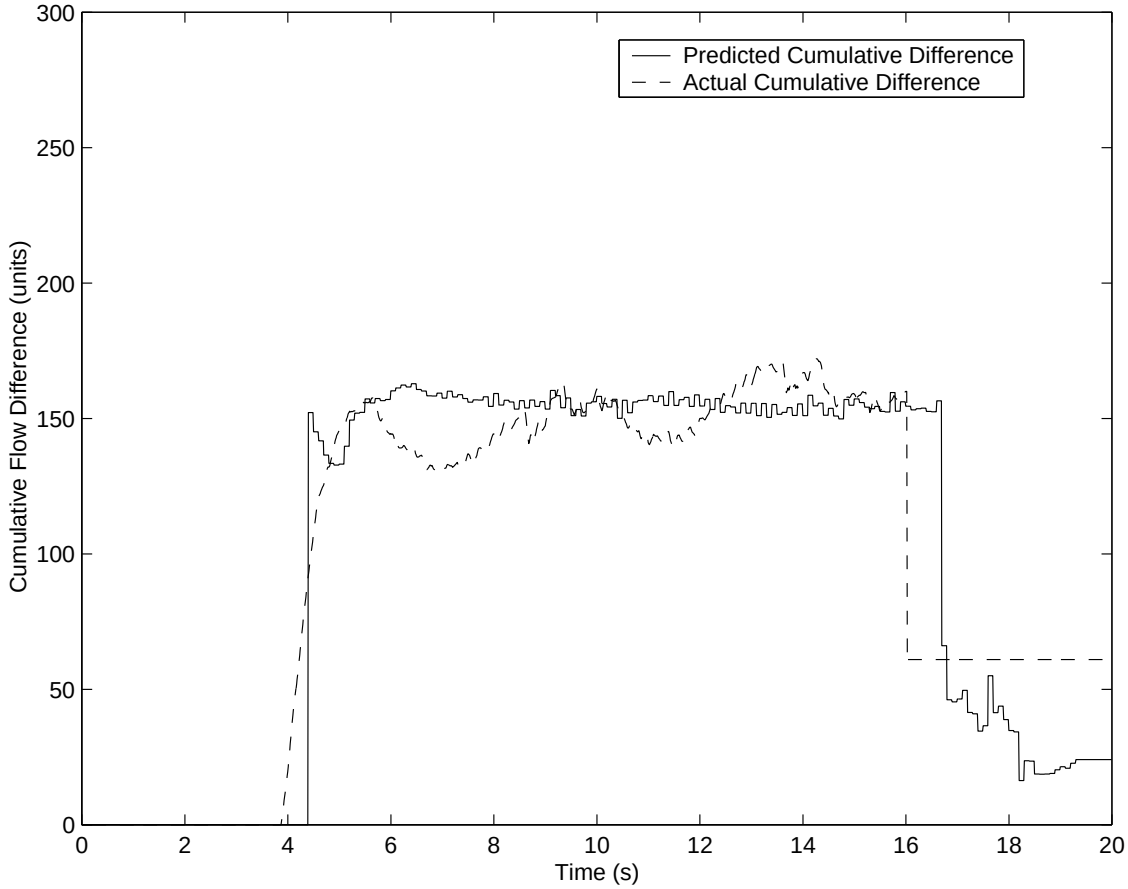

Fig. 56. 20-Step-Ahead Prediction of Cumulative Flow Difference between Sending and Arrival Rate Using Cross-Traffic 1.

objective. Controller 2 and Controller 3 equations, include multiple terms in order to accomplish all three objectives. In order for the MPC technique to be beneficial at least two predictors should be used working simultaneously. One should keep the buffer level above the minimum while the second would be achieving loss minimization. A technique of developing a control equation for multi-input multi-output systems could also be applicable.

6. Importance of the Initial Source Buffering

It would be quite enlightening, if the relation between the initial source buffer level and the system dead-time was demonstrated. Therefore, the effects of a limited initial source buffering are evaluated. For this reason two simulations are run with Controller 4 under network propagation delay of 0.05 seconds. The initial source buffering is reduced by 30% as compared to the case shown in Figures 57 and 58. Figures 59, 60, 61 and 62 show that when the source buffering is limited, the resulting control effort is essentially the same as in the open-loop. Because the initial source buffer level is lower in this simulation than in previous simulations, it is more likely to be emptied as the controller attempts to maintain the desired destination buffer. However, as the initial source buffering is decreased the system dead-time is minimized. If the source buffer is empty, then the output rate of the source buffer cannot be larger than the input rate to the source buffer, which is assumed to be constant. Therefore, even though a larger control effort is desired, only the constant source rate is available. The result is that the system reacts much like the open-loop case and approaches open-loop performance as the initial source buffer level approaches zero.

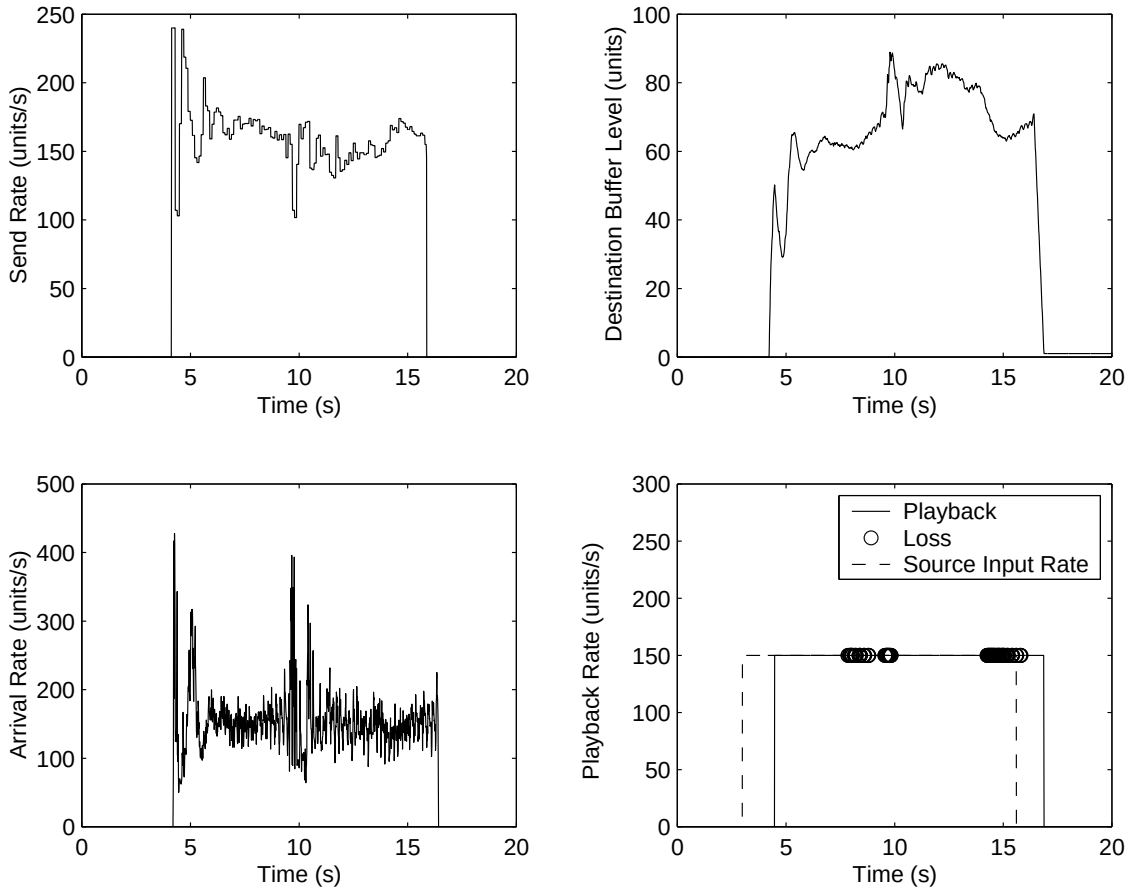

Fig. 57. Buffer Level and Flow Rates for MPC Simulation Using Cross-Traffic 2; Destination Buffer Regulation.

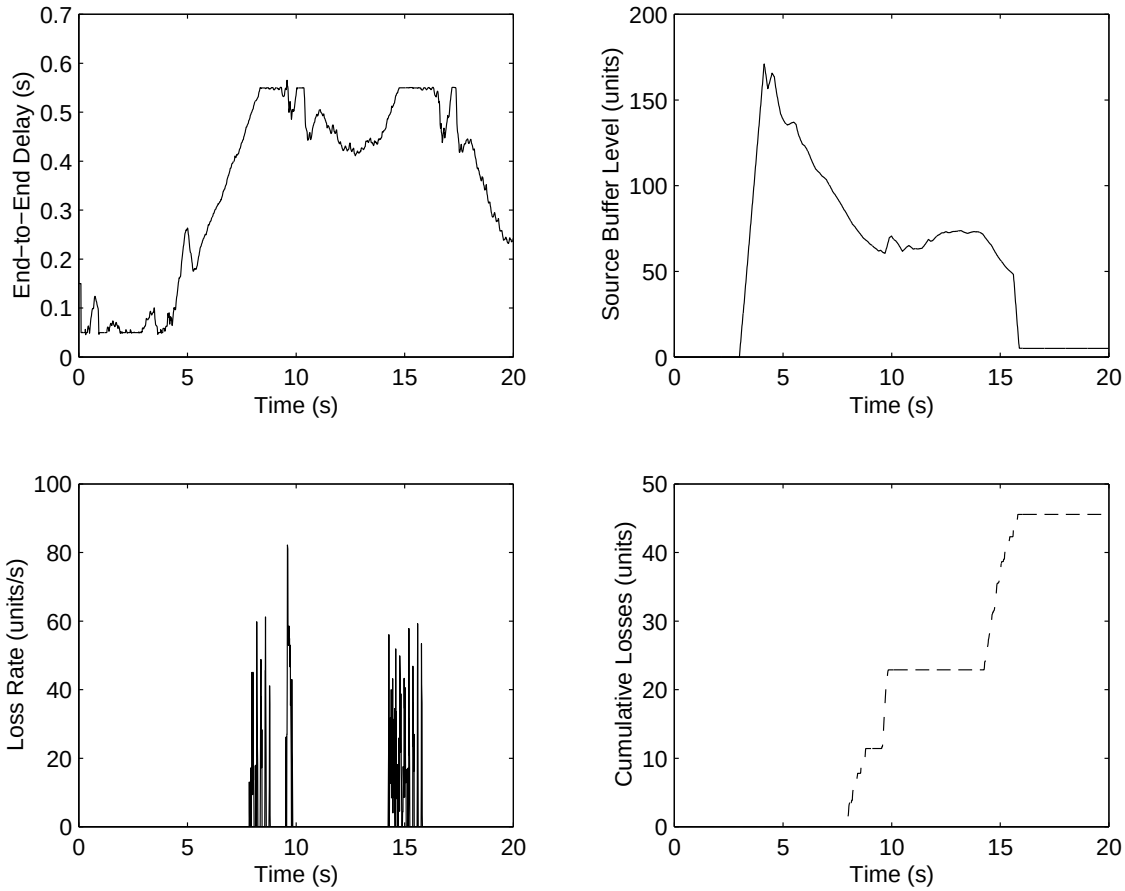

Fig. 58. End-to-End Delay, Source Buffer Level and Losses for MPC Simulation Using Cross-Traffic 2; Destination Buffer Regulation.

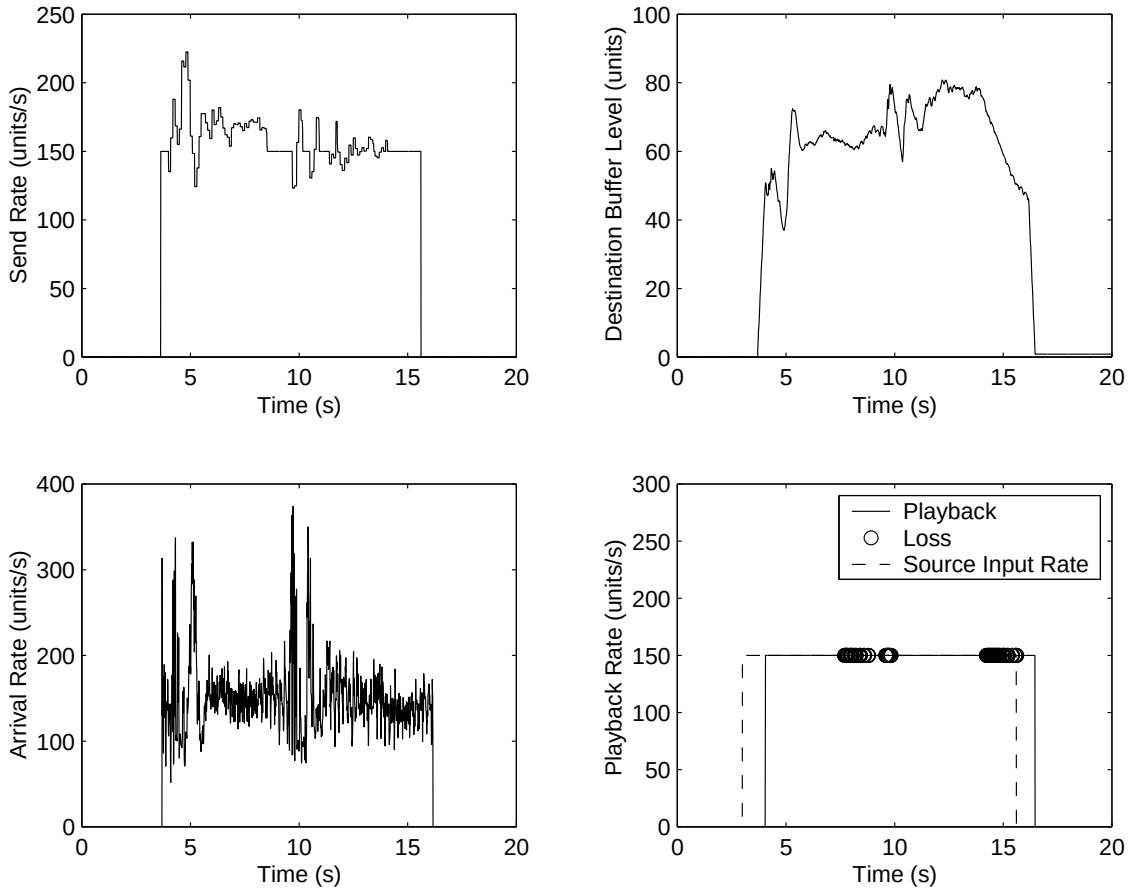

Fig. 59. Buffer Level and Flow Rates for MPC Simulation Using Cross-Traffic 2 for 30% Decreased Initial Source Buffering; Destination Buffer Regulation.

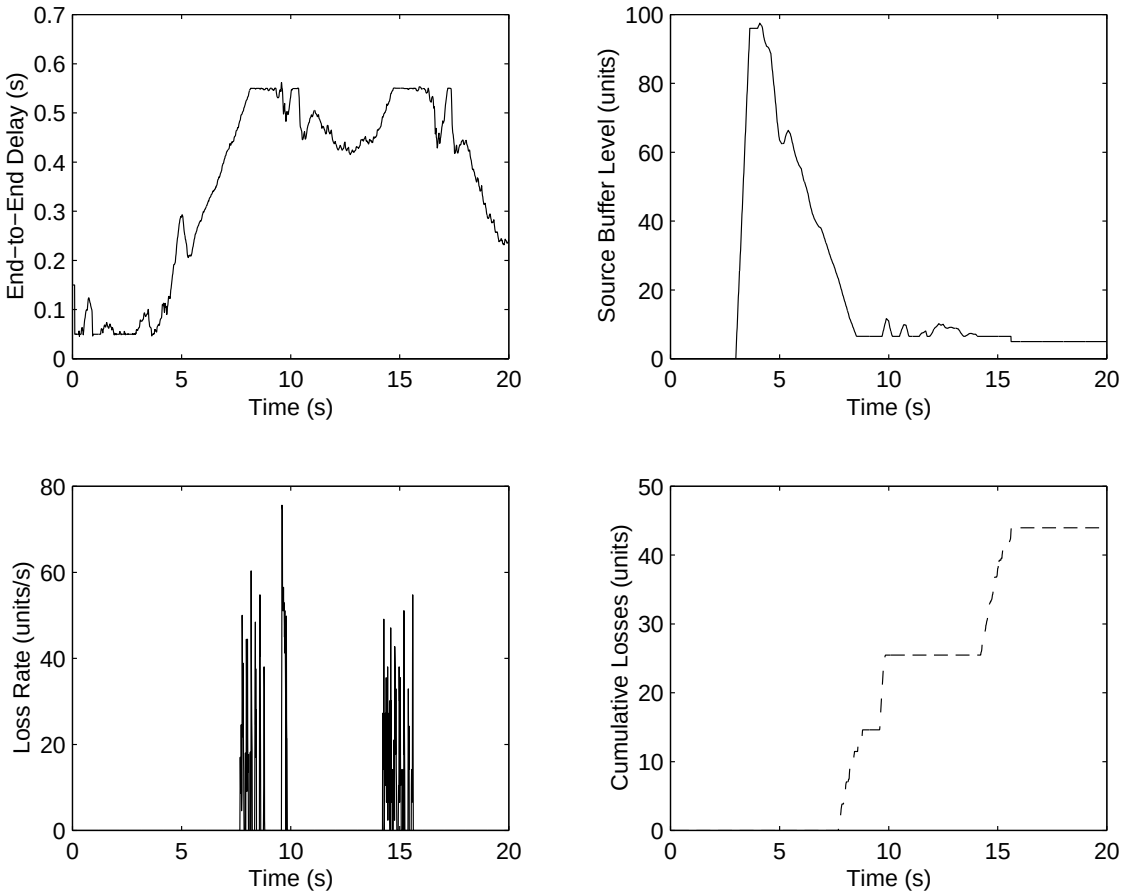

Fig. 60. End-to-End Delay, Source Buffer Level and Losses for MPC Simulation Using Cross-Traffic 2 for 30% Decreased Initial Source Buffering; Destination Buffer Regulation.

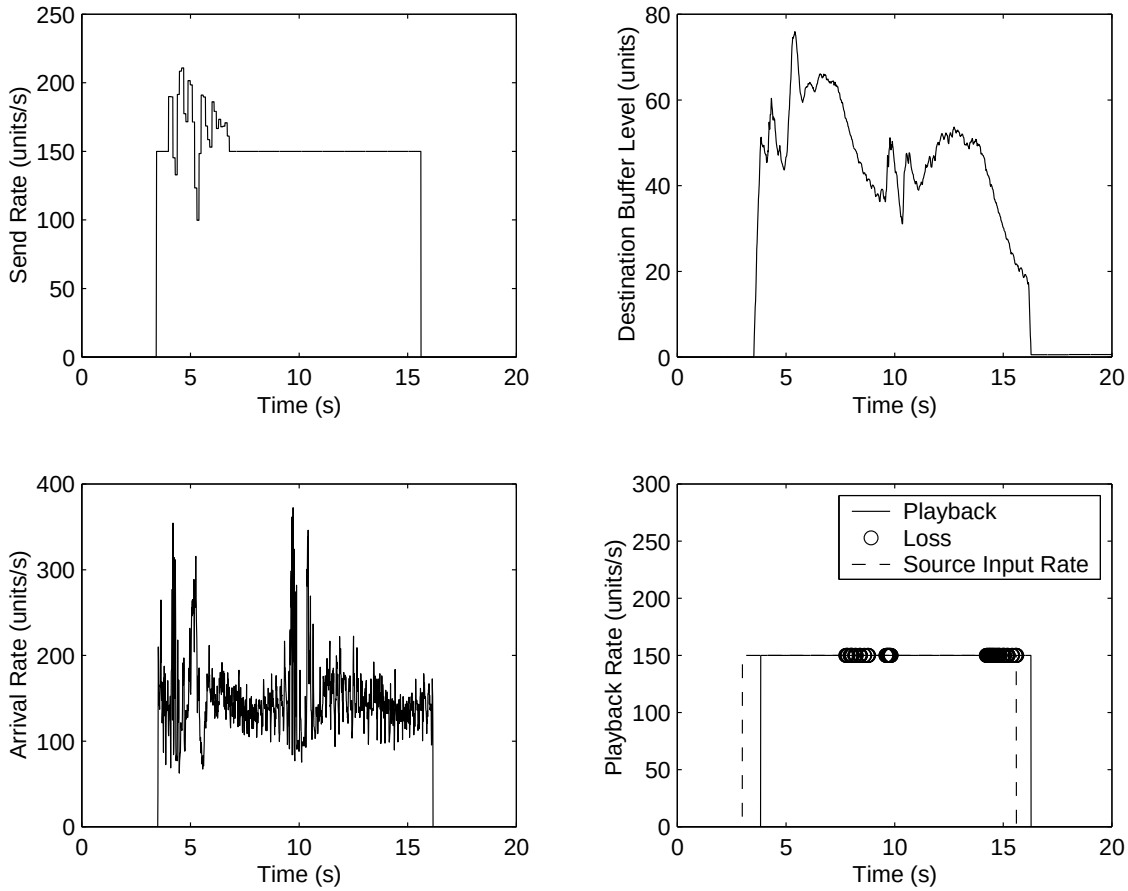

Fig. 61. Buffer Level and Flow Rates for MPC Simulation Using Cross-Traffic 2 for 65% Decreased Initial Source Buffering; Destination Buffer Regulation.

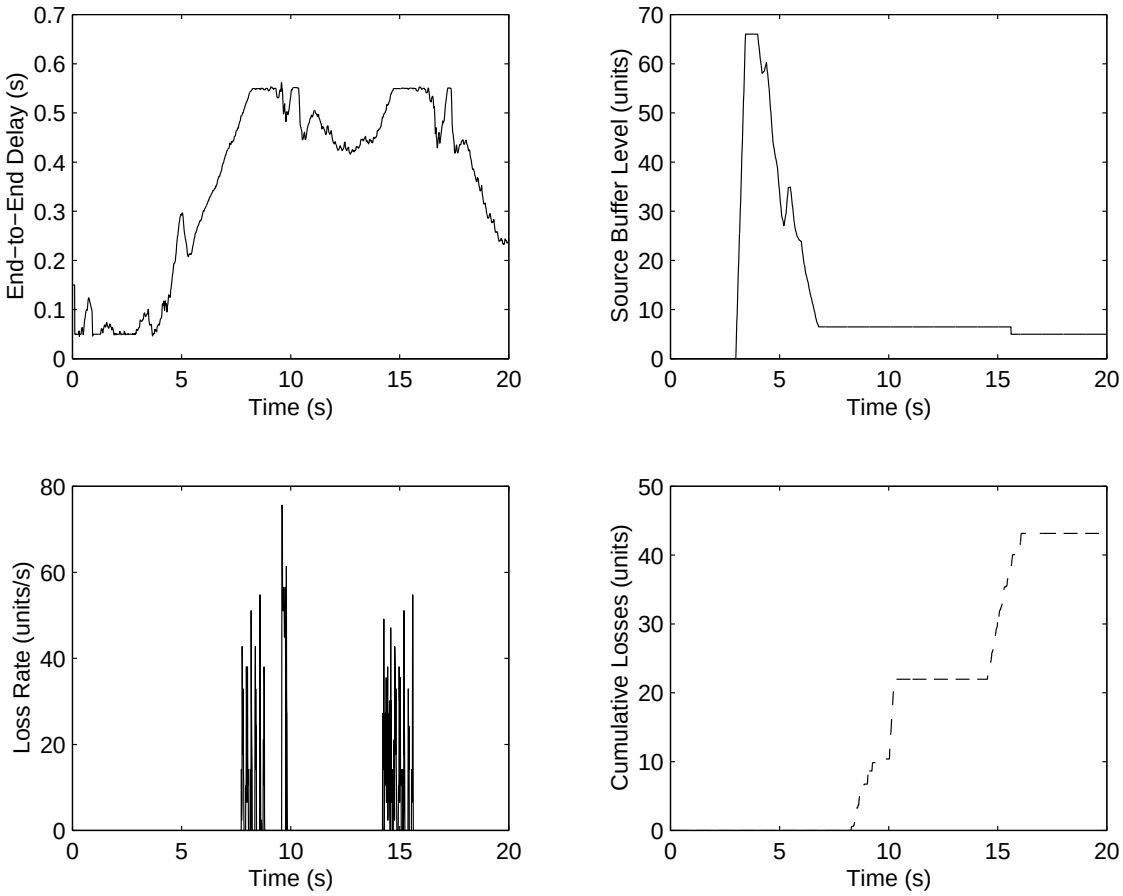

Fig. 62. End-to-End Delay, Source Buffer Level and Losses for MPC Simulation Using Cross-Traffic 2 for 65% Decreased Initial Source Buffering; Destination Buffer Regulation.

C. Effectiveness of Flow Controllers for 20% Decrease in Application Send Rate

In this section the controller performance is investigated, when the application send rate is decreased. In the previous simulations the send rate is 150 ups, whereas in the simulations shown in this section the send rate is 120 ups. Controller 2 and 3 are tested under this condition. Comparison basis is the open-loop simulation for the same send rate.

1. Open-Loop Simulation

Figures 63, 64 and 65 show the system variables for the uncontrolled case using cross-traffic trace 1. Since the send rate is decreased the losses are also decreased. The dead-time for this simulation is 1.45 seconds and there are 40 losses.

2. Reactive Implementation of Controller 2

For Controller 2 the system response is shown in Figures 66 and 67. Even though implemented reactively Controller 2 is able to impact the losses. A 33% decrease on the losses is the result. The tradeoff is a 20% increase in the dead-time. The effect of the inverse delay term is obvious. The controller inverts the effects of the time-varying time-delay effectively.

Although the controller is implemented reactively the results indicate a significant improvement on the losses and therefore to the throughput achieved. This of course is justified by the slow change of the delay as earlier mentioned.

3. Predictive Implementation of Controller 2

For Controller 2 the system response is shown when the it is implemented predictively. The losses are reduced by 45%. The dead-time is increased by 23%. The results are

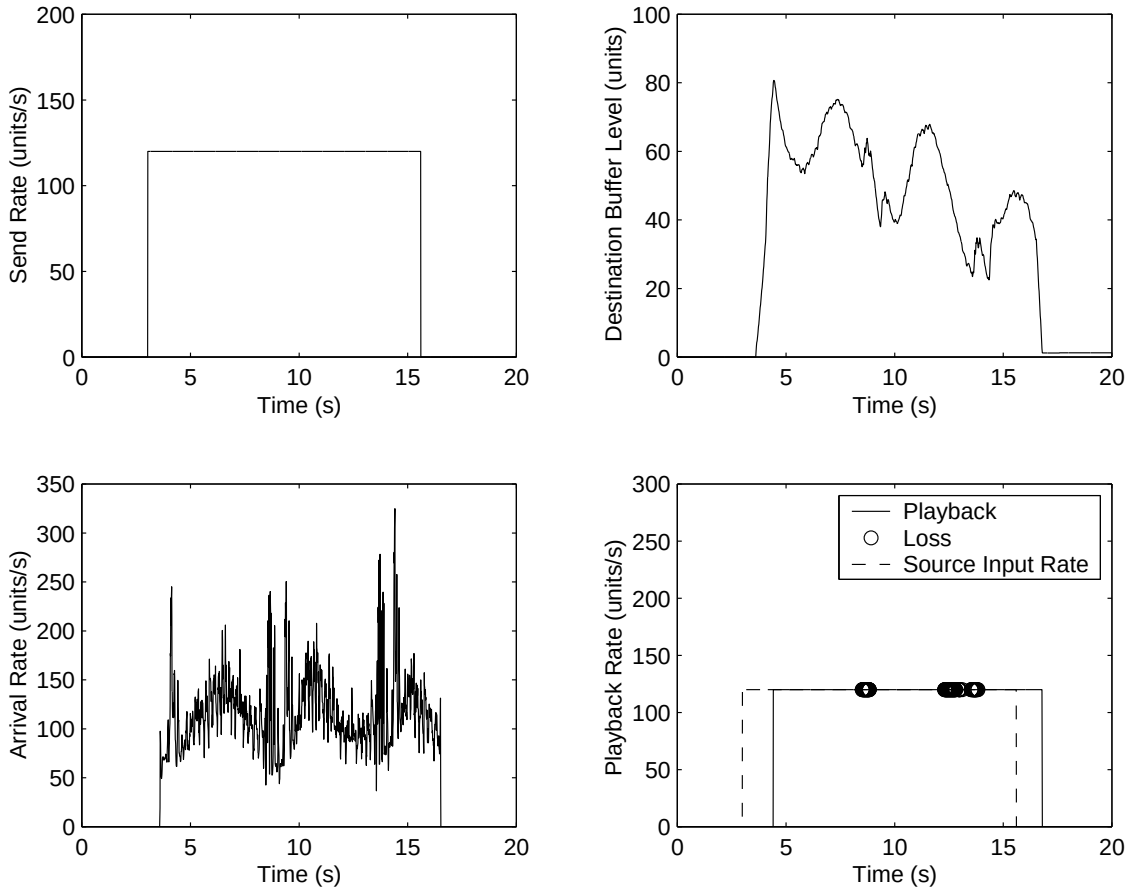

Fig. 63. Buffer Level and Flow Rates for Open-Loop Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups.

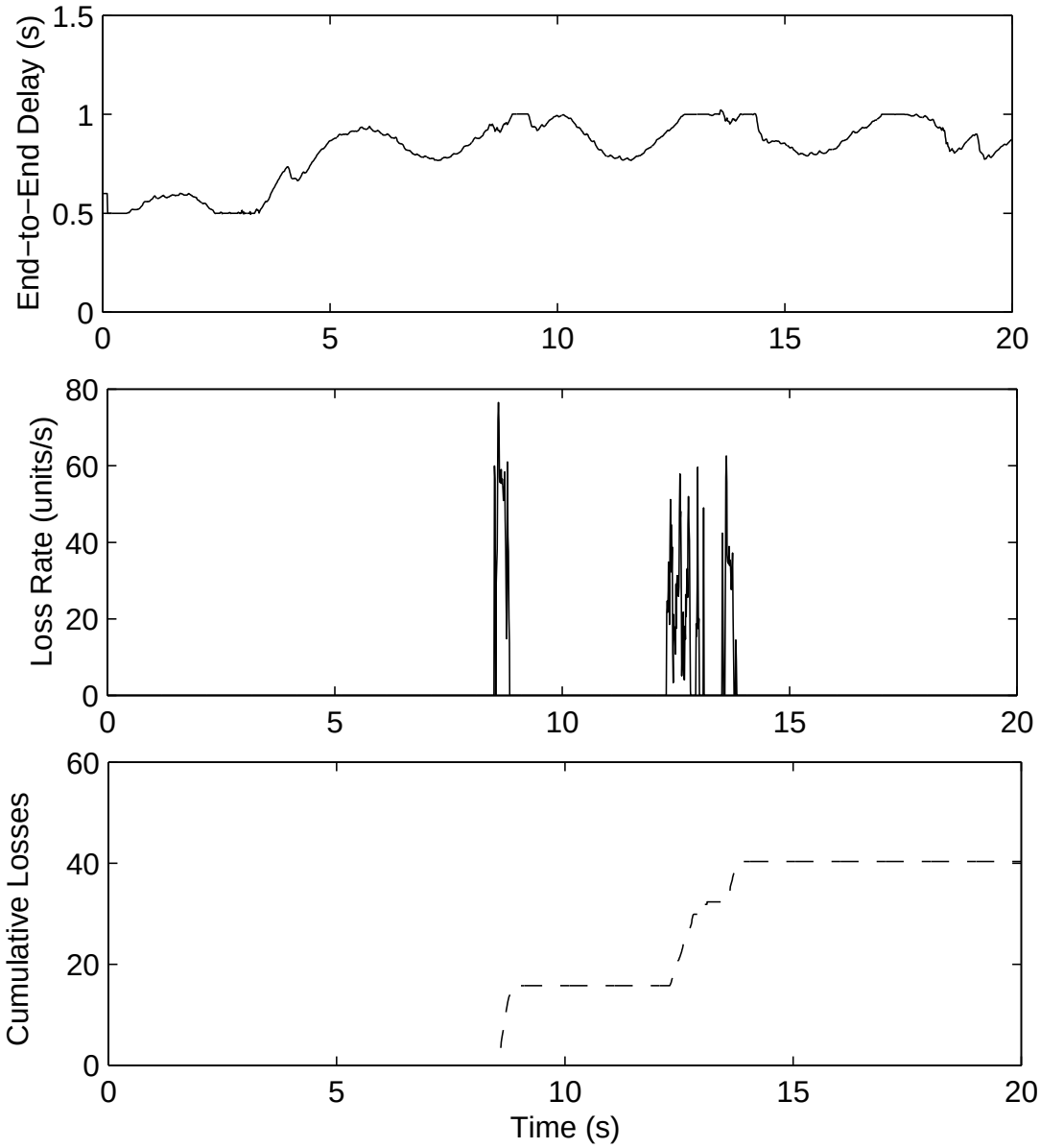

Fig. 64. End-to-End Delay and Losses for Open-Loop Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups.

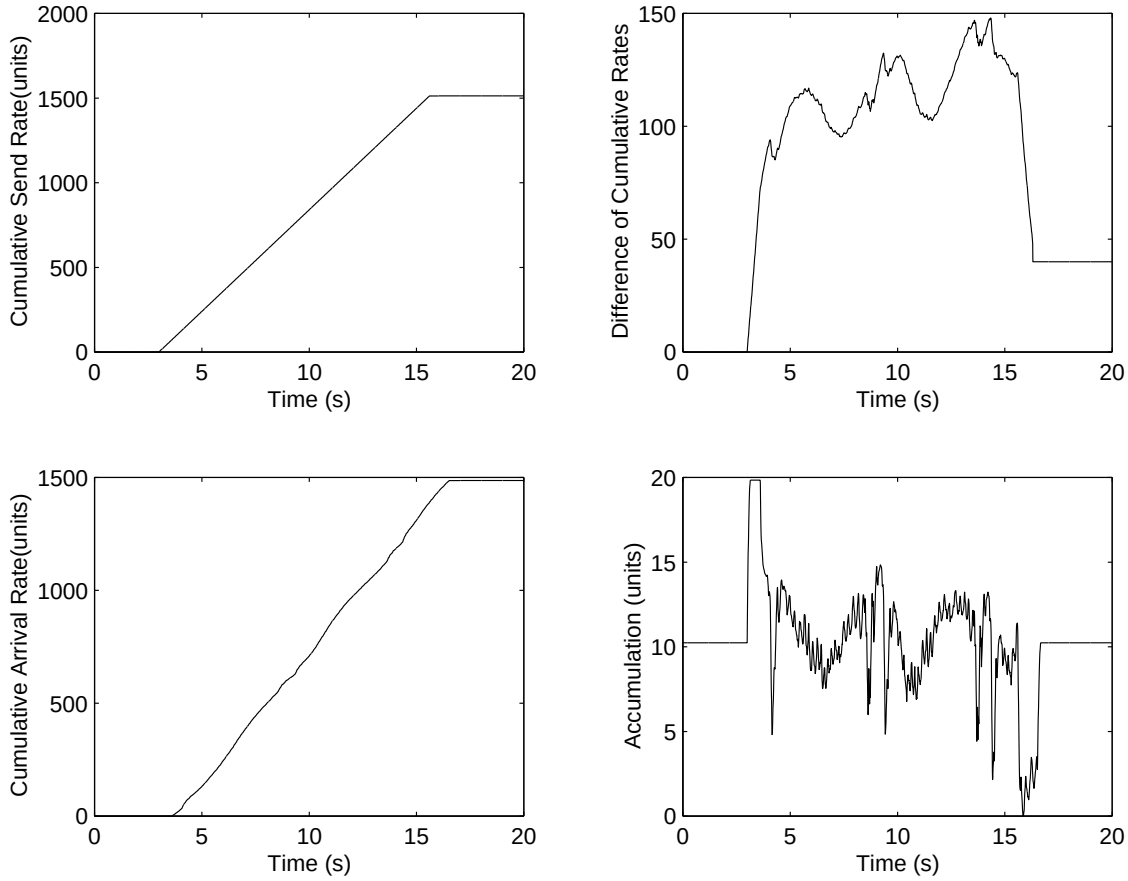

Fig. 65. Cumulative Flow Rates and Accumulation for Open-Loop Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups.

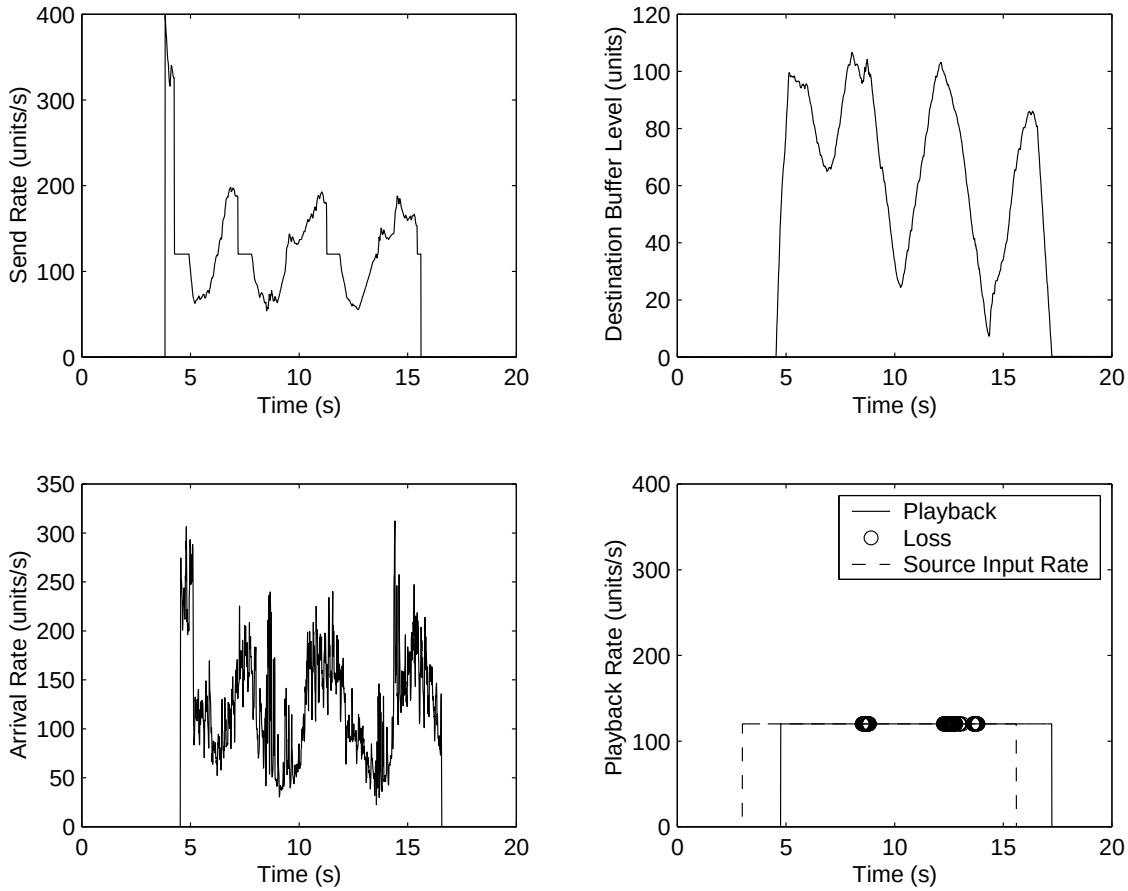

Fig. 66. Buffer Level and Flow Rates for Reactive Controller 2 Using Cross-Traffic 1 for Application Send Rate of 120 ups.

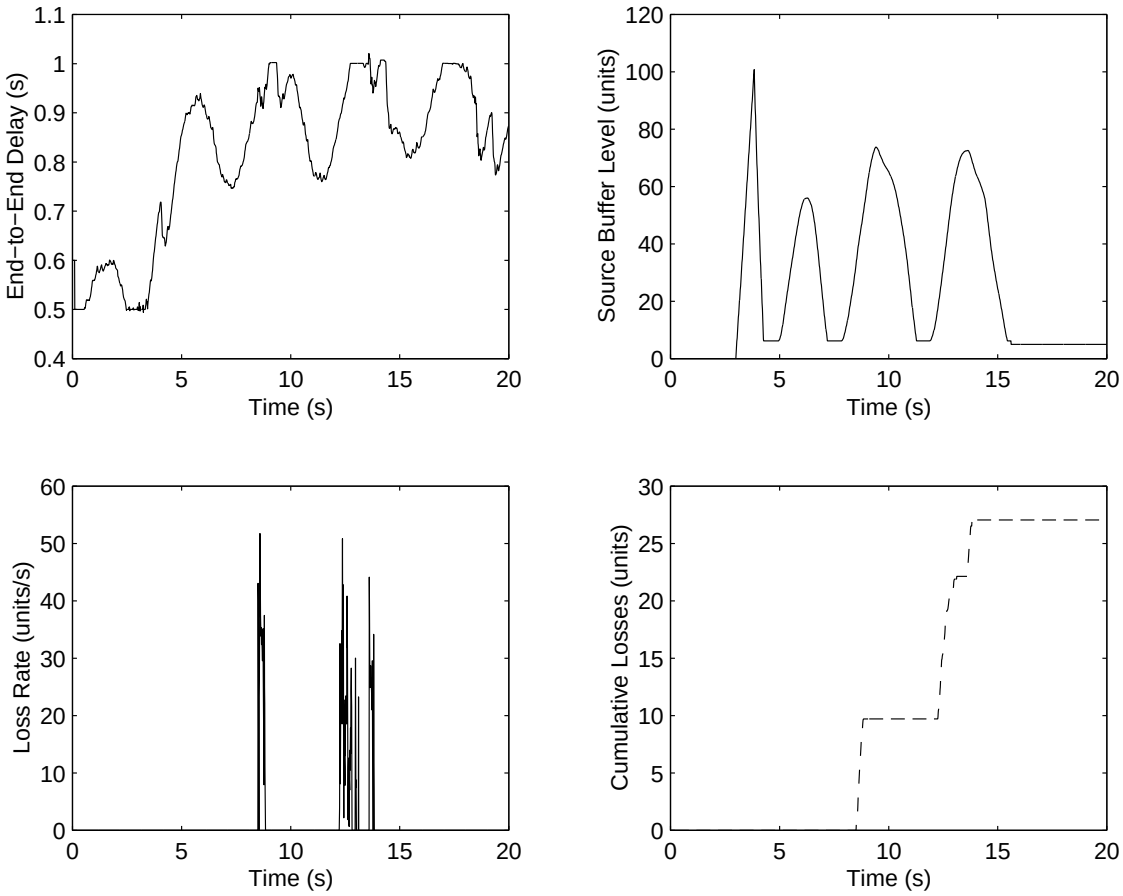

Fig. 67. End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups.

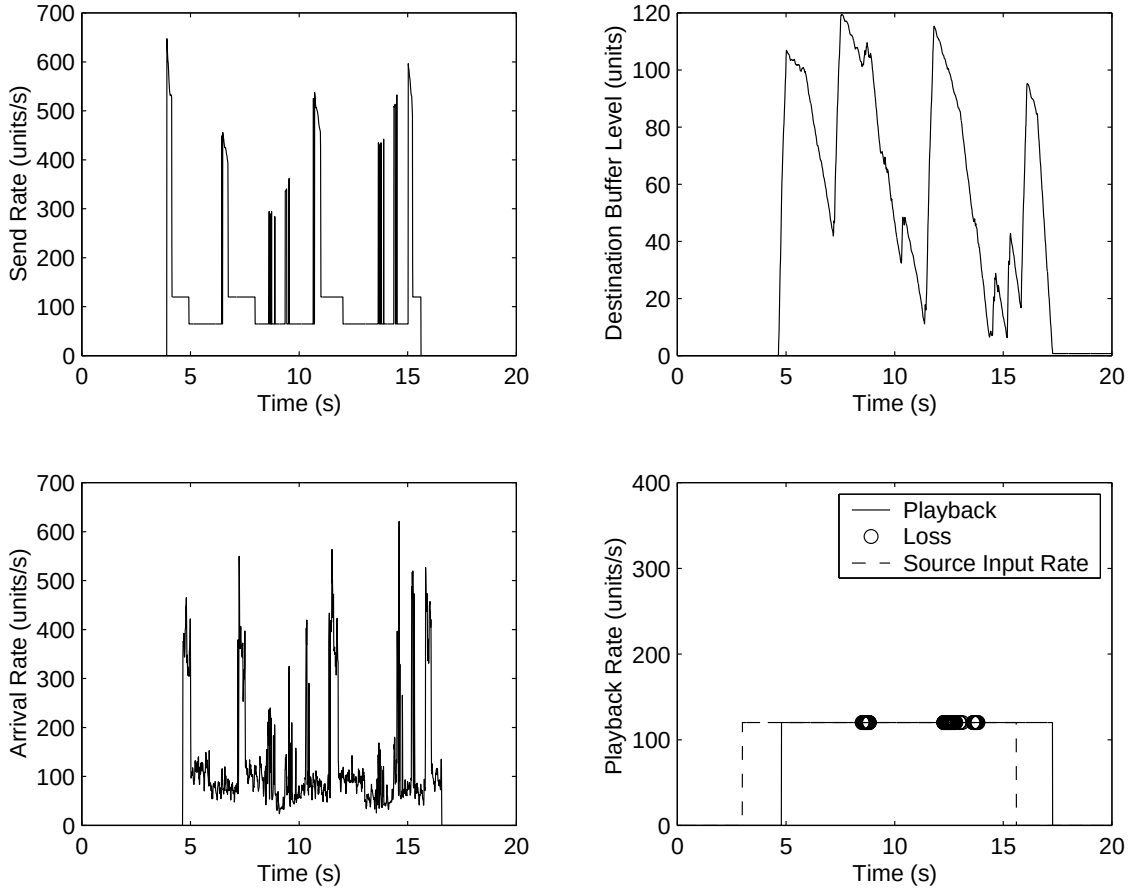

Fig. 68. Buffer Level and Flow Rates for Predictive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups.

shown in Figures 68 and 69.

4. Reactive Implementation of Controller 3

The system response for this case is shown in Figures 70, 71 and 72. Controller 3 achieves a 20% decrease in the losses whereas the system dead-time is increased by 17%.

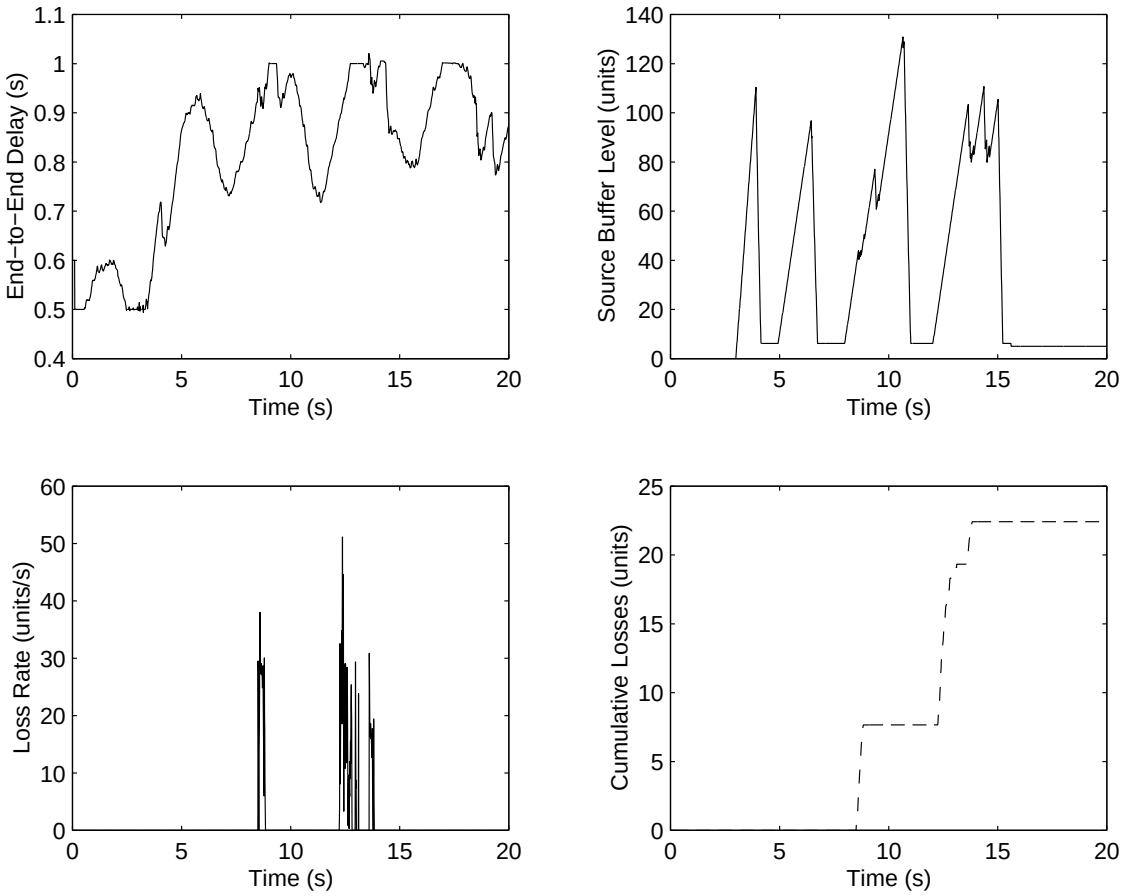

Fig. 69. End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups.

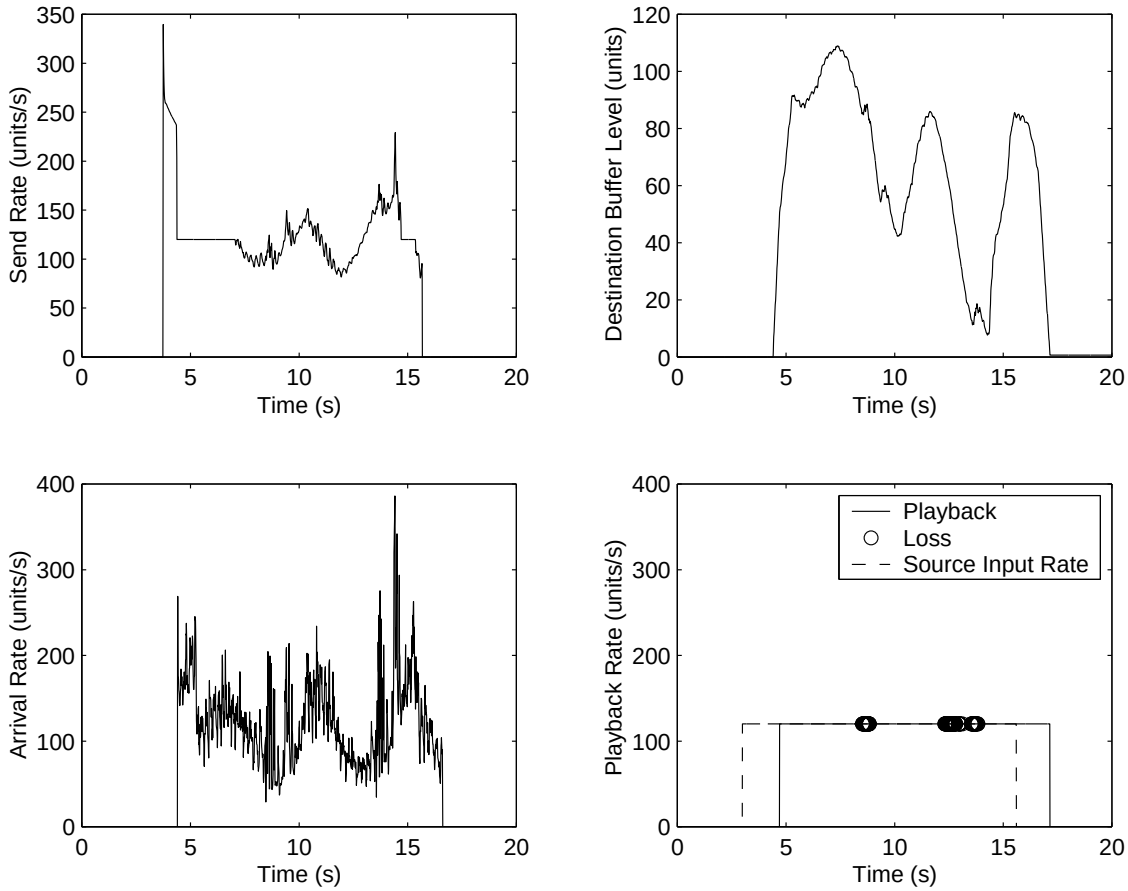

Fig. 70. Buffer Level and Flow Rates for Reactive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups.

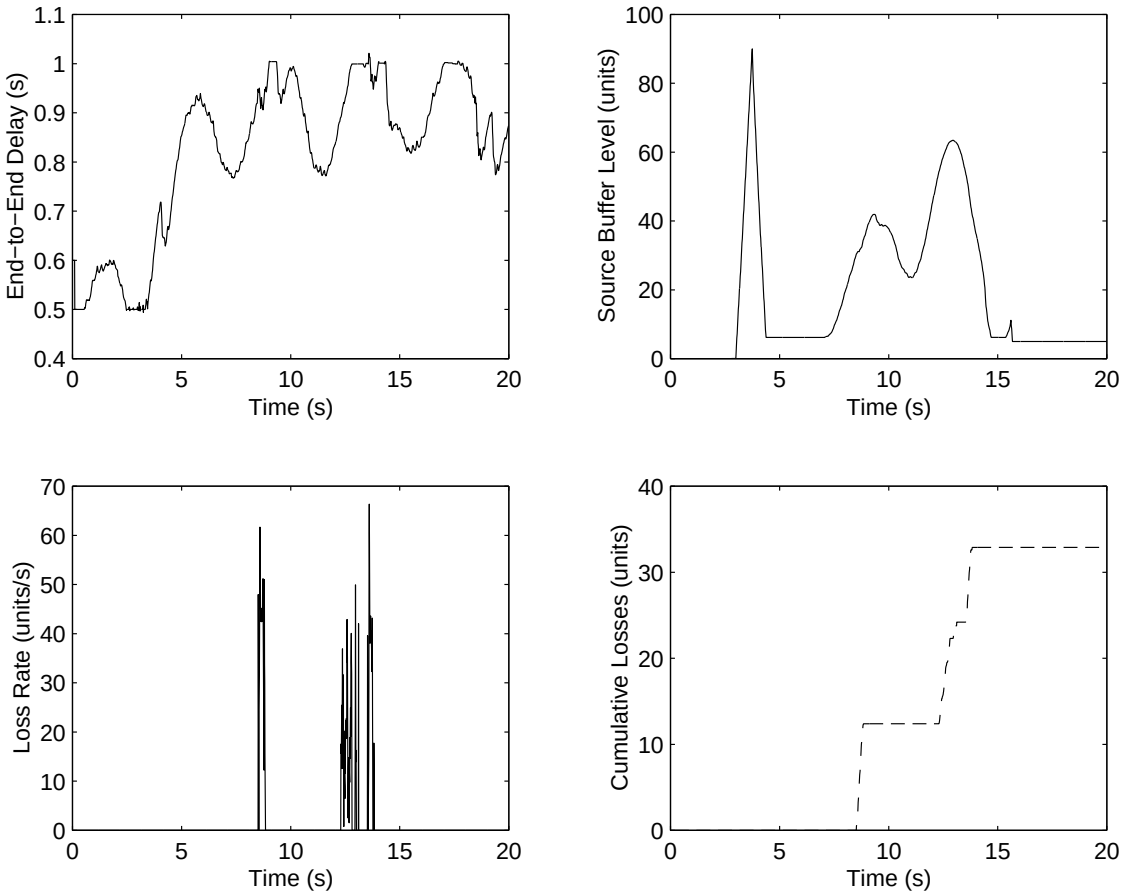

Fig. 71. End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups.

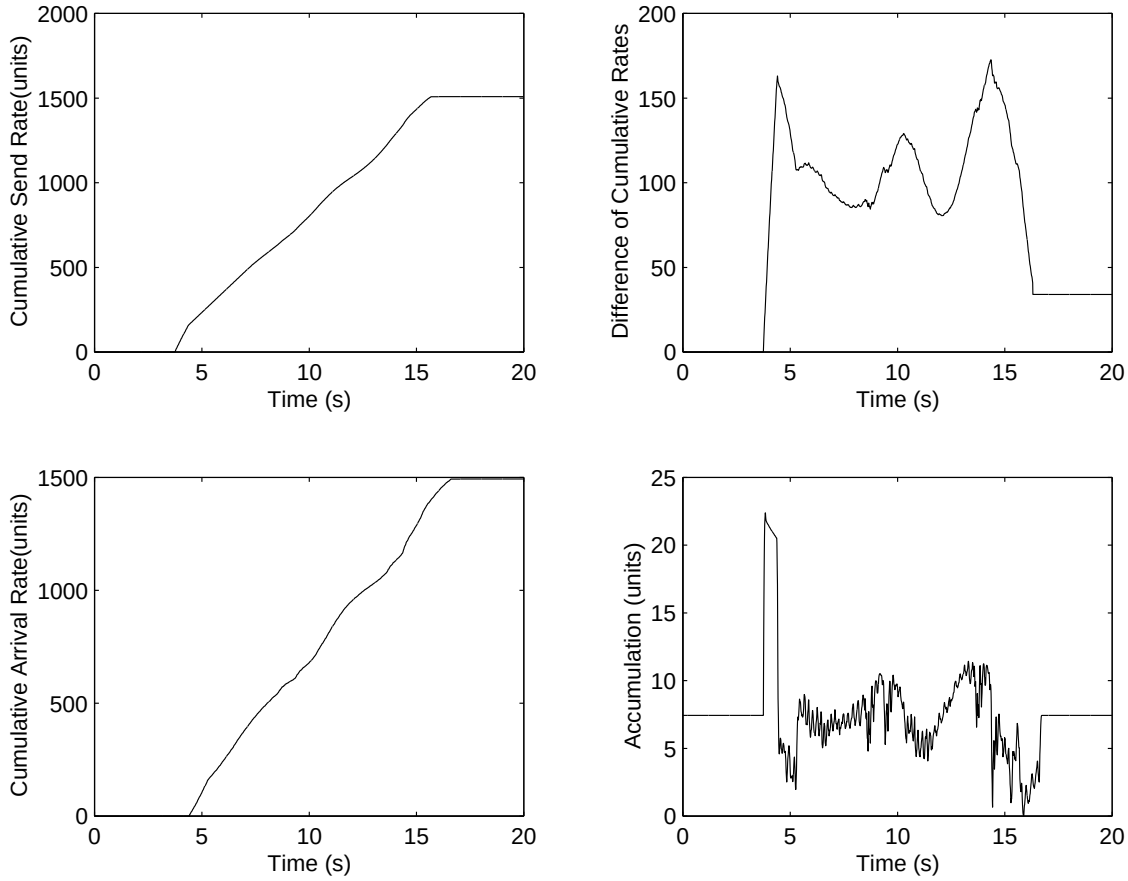

Fig. 72. Cumulative Flow Rates and Accumulation for Reactive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups.

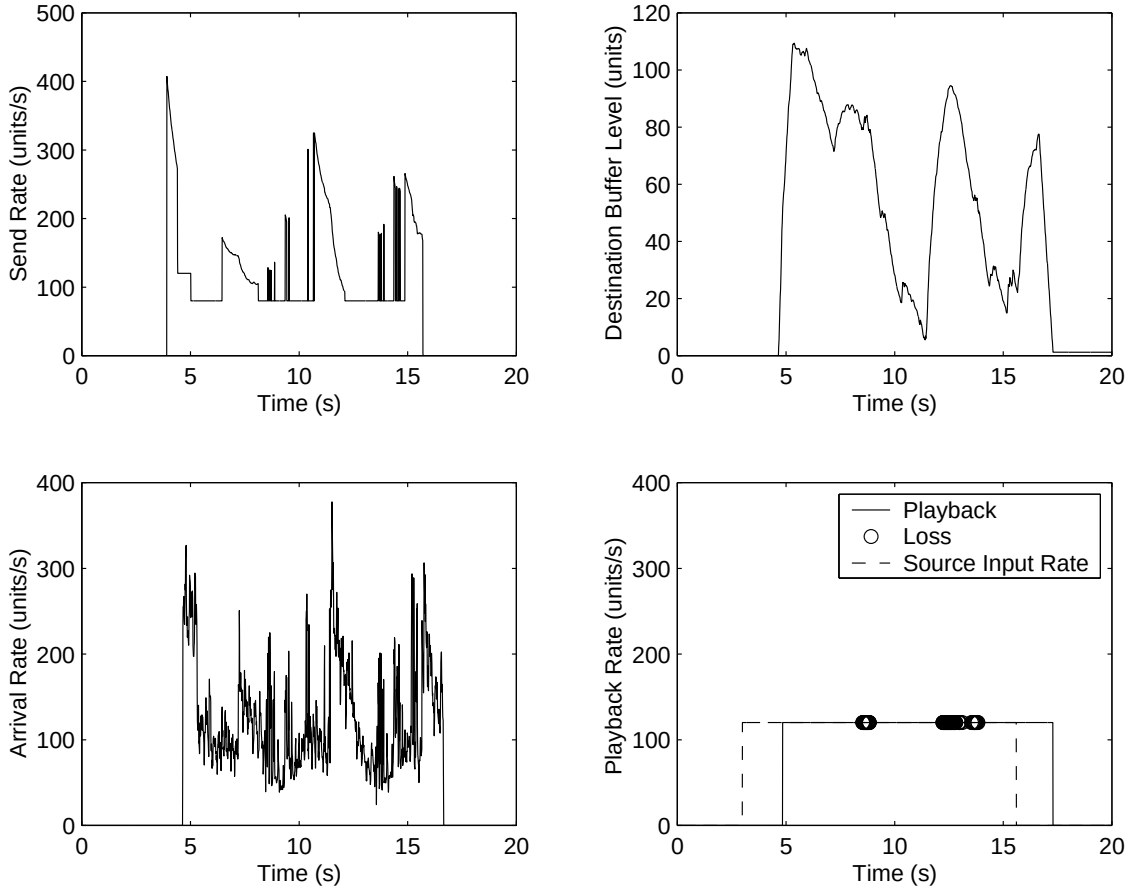

Fig. 73. Buffer Level and Flow Rates for Cross-Traffic 1 for Predictive Controller 3 Simulation Using Application Send Rate of 120 ups.

5. Predictive Implementation of Controller 3

Figures 73, 74 and 75 show the simulation results of the predictive implemented Controller 3. It results in a 40% decrease in the losses. This is accomplished by a 23% increase in the system dead-time.

The use of future information is proved to be advantageous as the predictor warns the controller to reduce the send rate when a delay spike is expected.

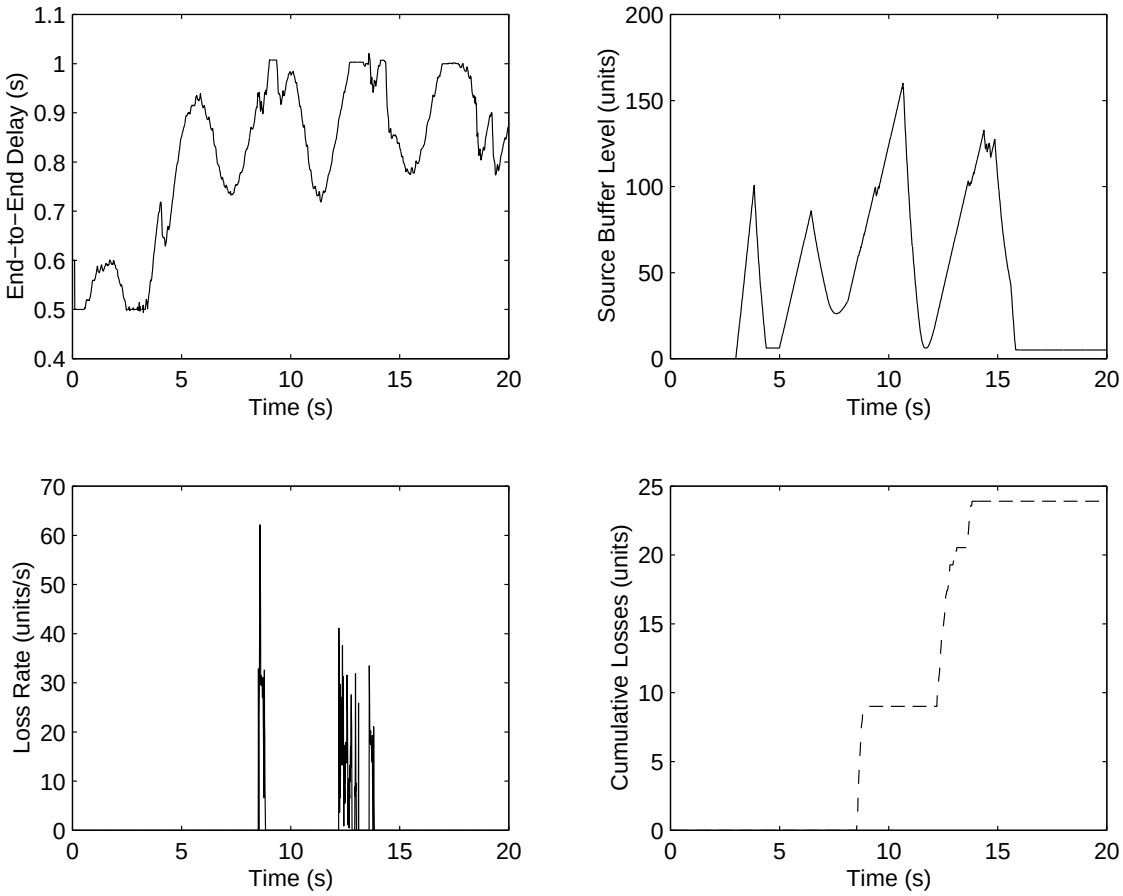

Fig. 74. End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups.

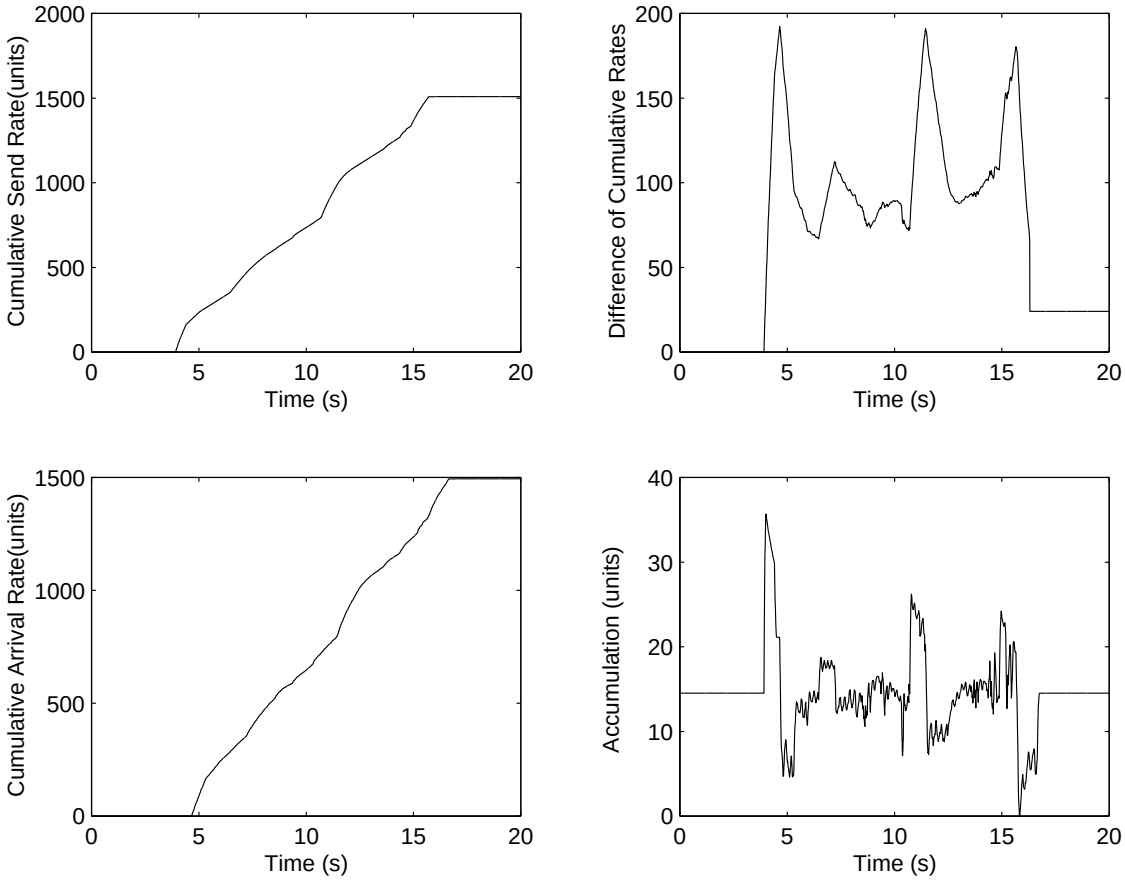

Fig. 75. Cumulative Flow Rates and Accumulation for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 120 ups.

D. Effectiveness of Flow Controllers for 20% Increase in Application Send Rate

In this section the controller performance is investigated, when the application send rate is 180 ups. Increasing the send rate in the uncontrolled case results in increase of losses. Controllers 2 and 3 are again tested under this send rate conditions. Comparison basis is the open-loop simulation for the same send rate.

1. Open-Loop Simulation

Figures 76, 77 and 78 show the system response for the open-loop case using the cross-traffic trace 1. Observe the increase in the cumulative losses. The dead-time for this simulation is 1.58 seconds and there are 94 losses.

2. Reactive Implementation of Controller 2

The system response for Controller 2 simulation is shown in Figures 79 and 80. Controller 2 is able to impact the losses. A 13% decrease on the losses with a 3% increase in dead-time is achieved.

3. Predictive Implementation of Controller 2

The system response for Controller 2 simulation is shown when it is implemented predictively. Implementation of Controller 2 in a predictive manner results in 20% decrease on the losses. The dead-time is increased by 17% as shown in the Figures 81 and 82.

Keep in mind that the predictive controllers can be used to control the exact number of cumulative losses. If further decrease in the number of losses is desired, then the gain k_2 in equation (4.6) should be decreased. The desired result will come with the expense of playback disruptions. Figures 83 and 84 illustrate the importance

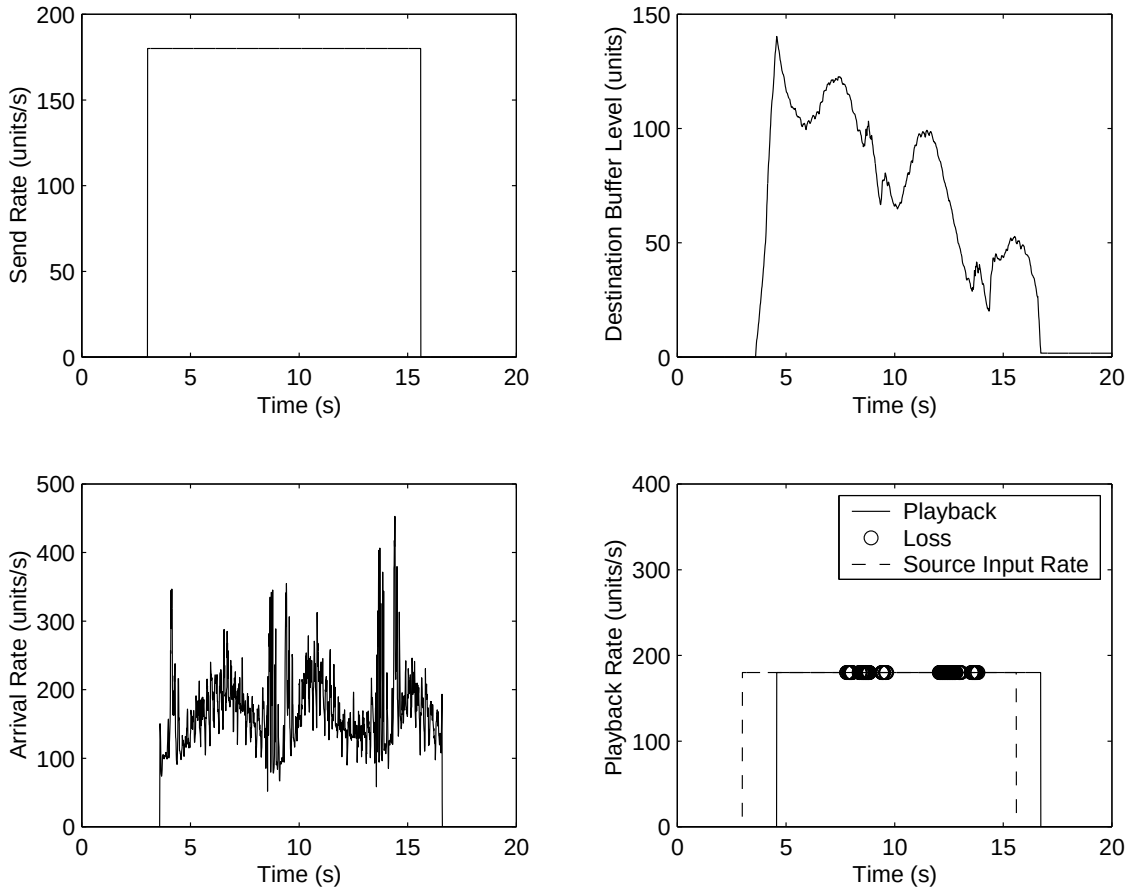

Fig. 76. Buffer Level and Flow Rates for Open-Loop Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups.

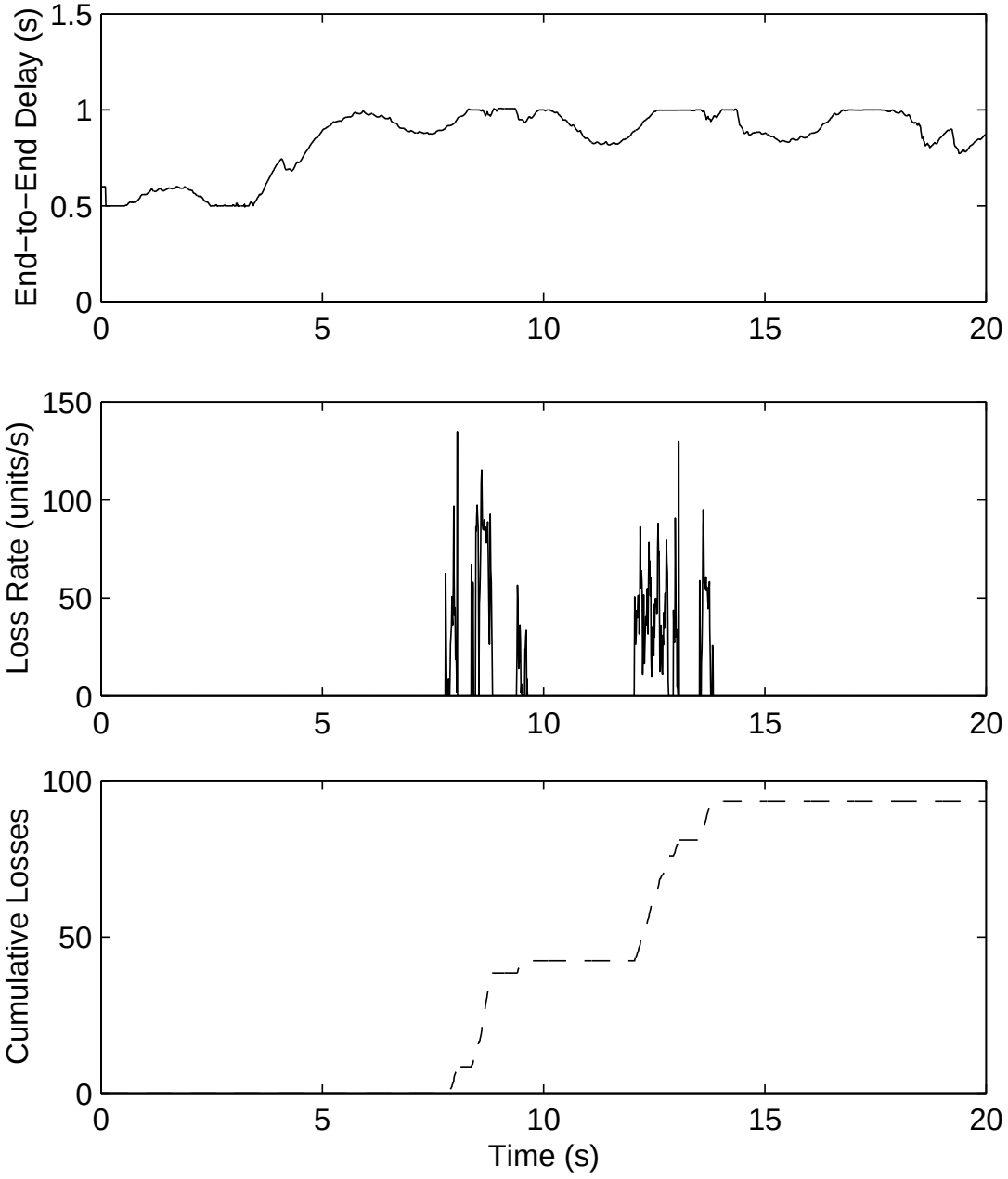

Fig. 77. End-to-End Delay and Losses for Open-Loop Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups.

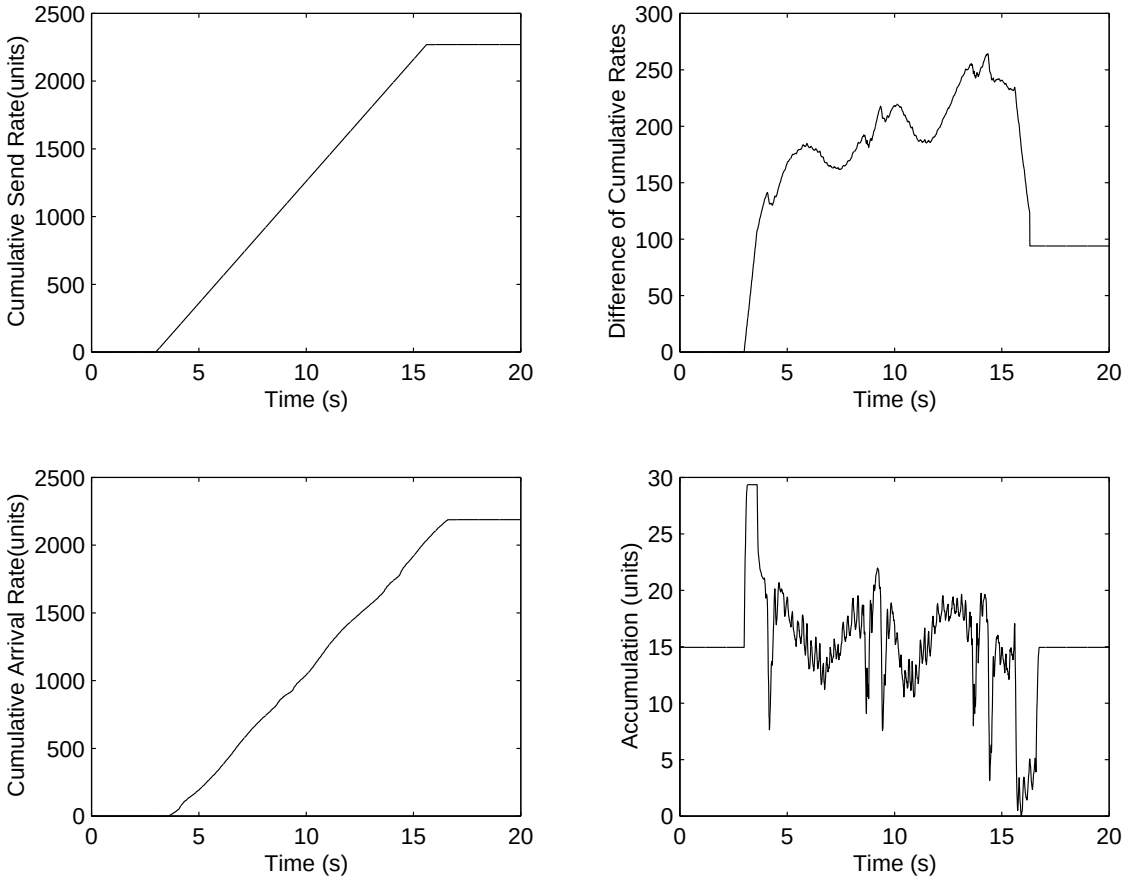

Fig. 78. Cumulative Flow Rates and Accumulation for Open-Loop Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups.

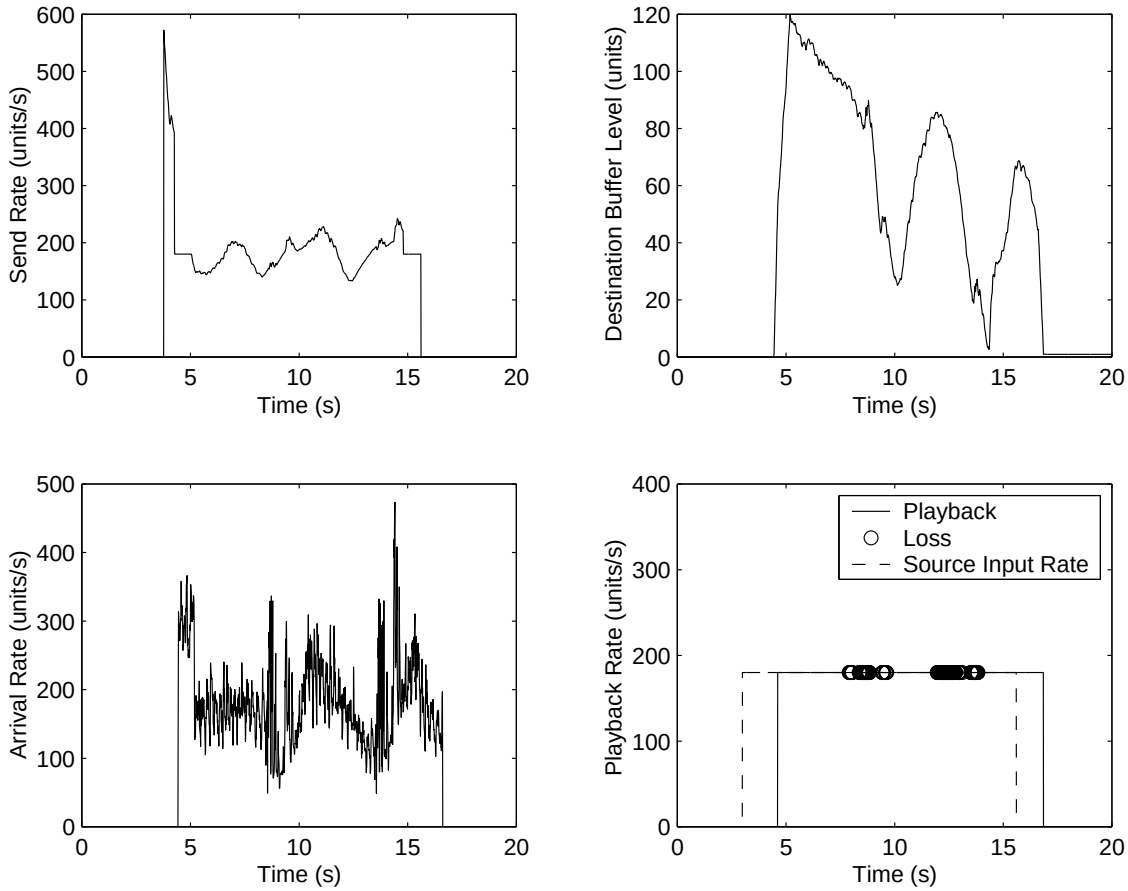

Fig. 79. Buffer Level and Flow Rates for Reactive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups.

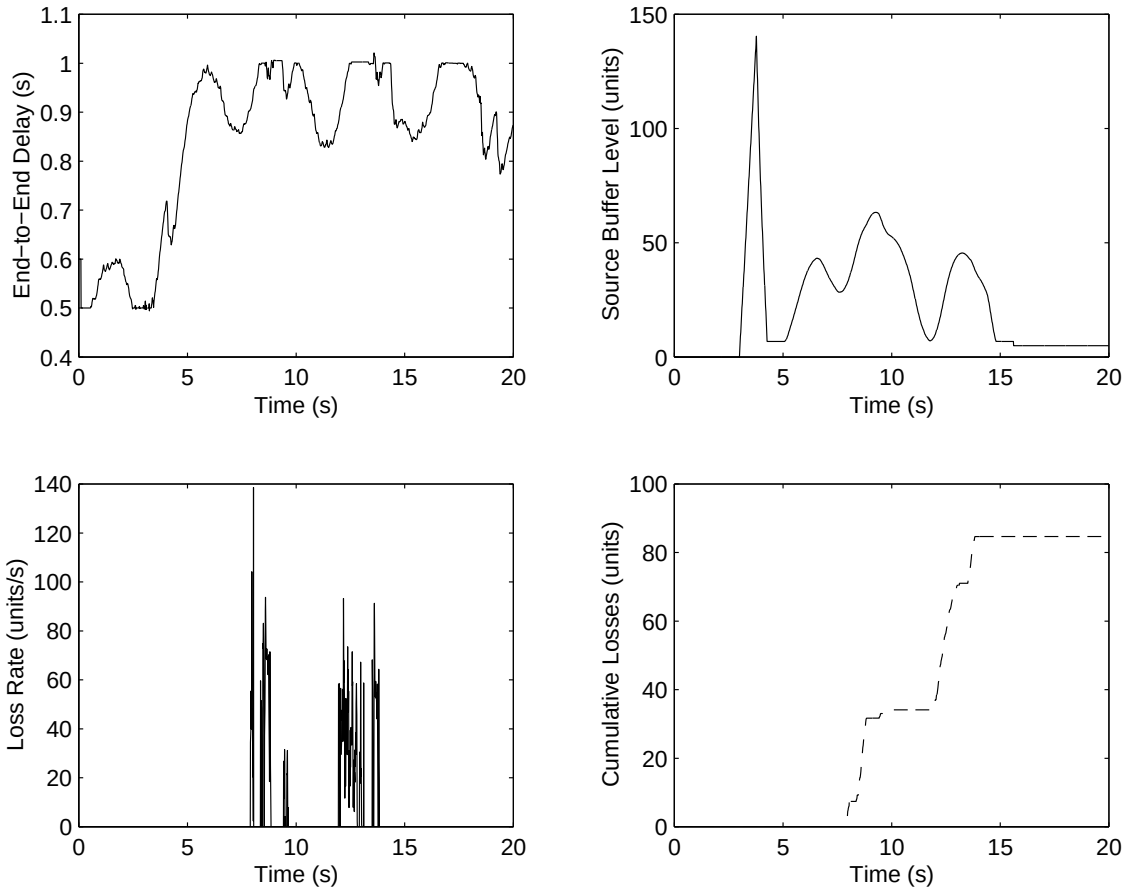

Fig. 80. End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups.

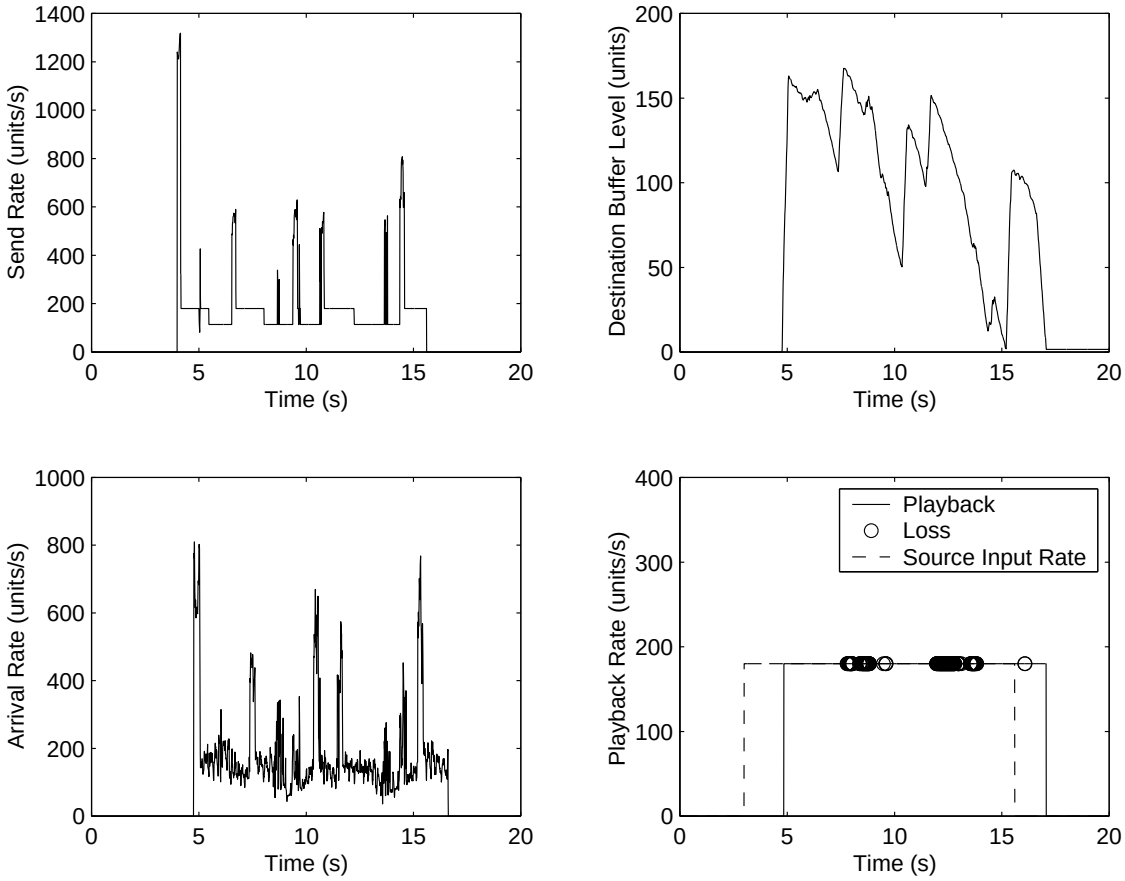

Fig. 81. Buffer Level and Flow Rates for Predictive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups.

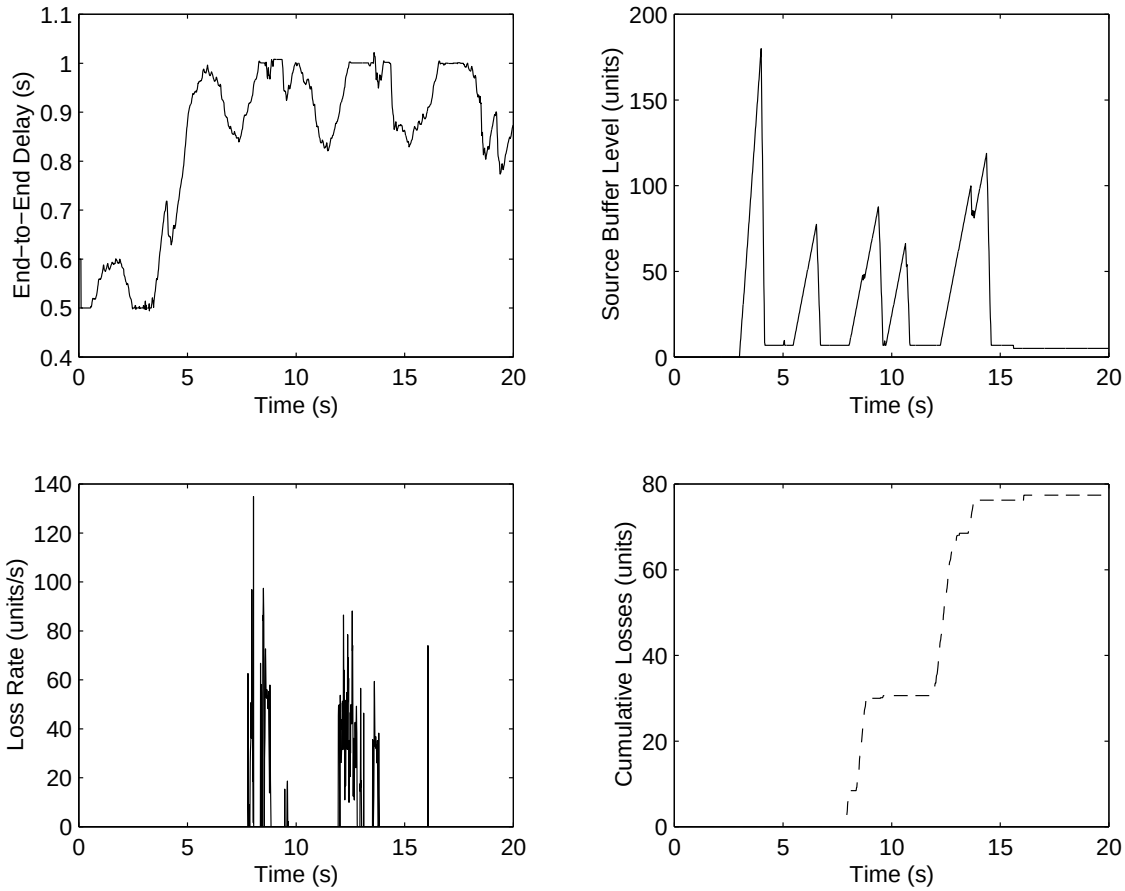

Fig. 82. End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups.

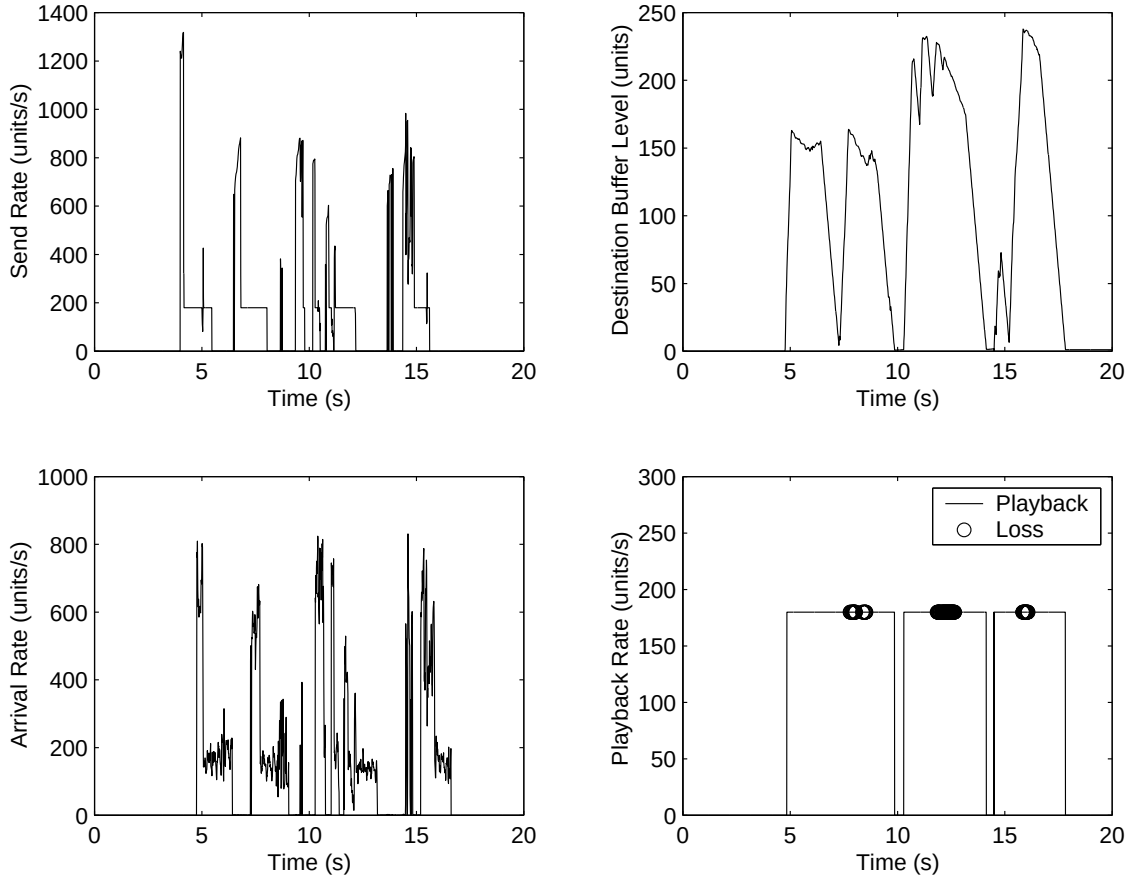

Fig. 83. Buffer Level and Flow Rates for Predictive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups and $k_2 = 0$.

of the gain k_2.

In Figures 83 and 84 show that for $k_2 = 0$ the losses are reduced by 42%. However, the playback as one can see is temporarily disrupted two times. The controller is shut off during periods of high end-to-end delay. In this way the losses are prevented. However, the destination buffer runs out of content during the periods of time the send rate is 0.

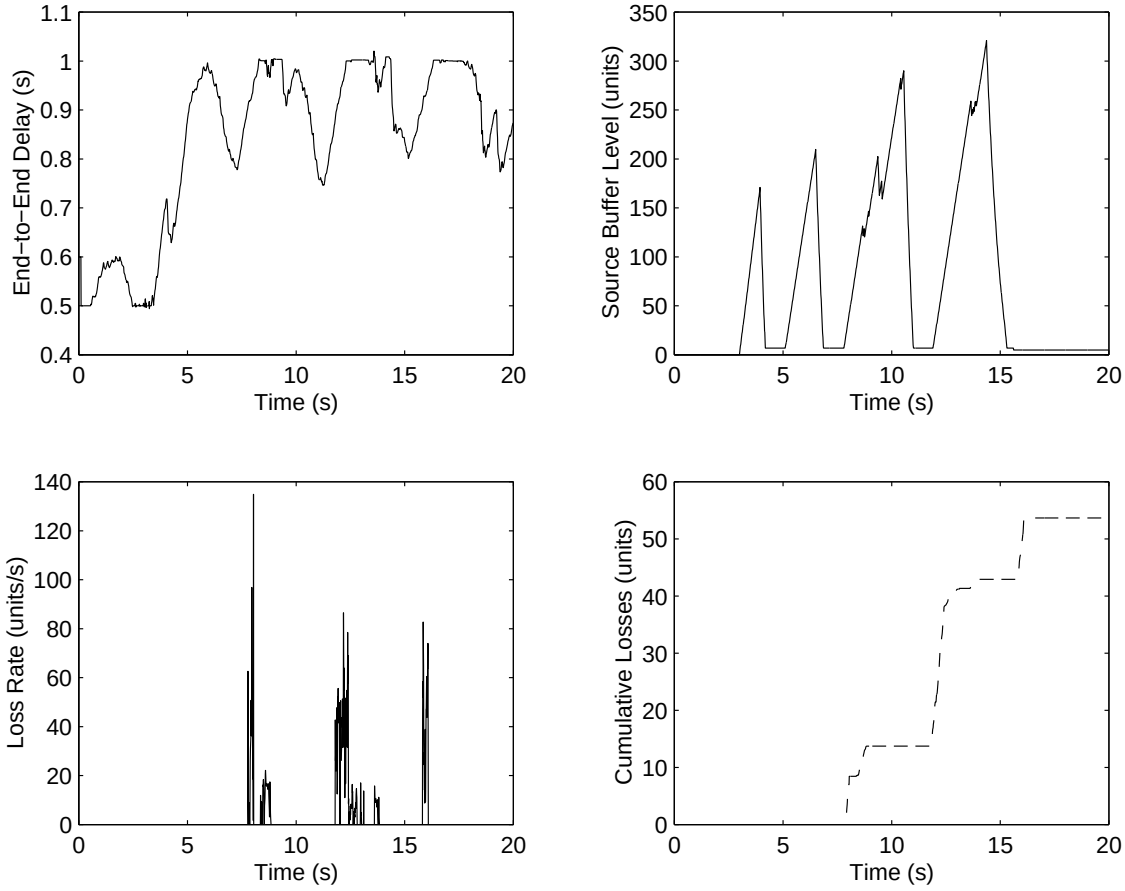

Fig. 84. End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 2 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups and $k_2 = 0$.

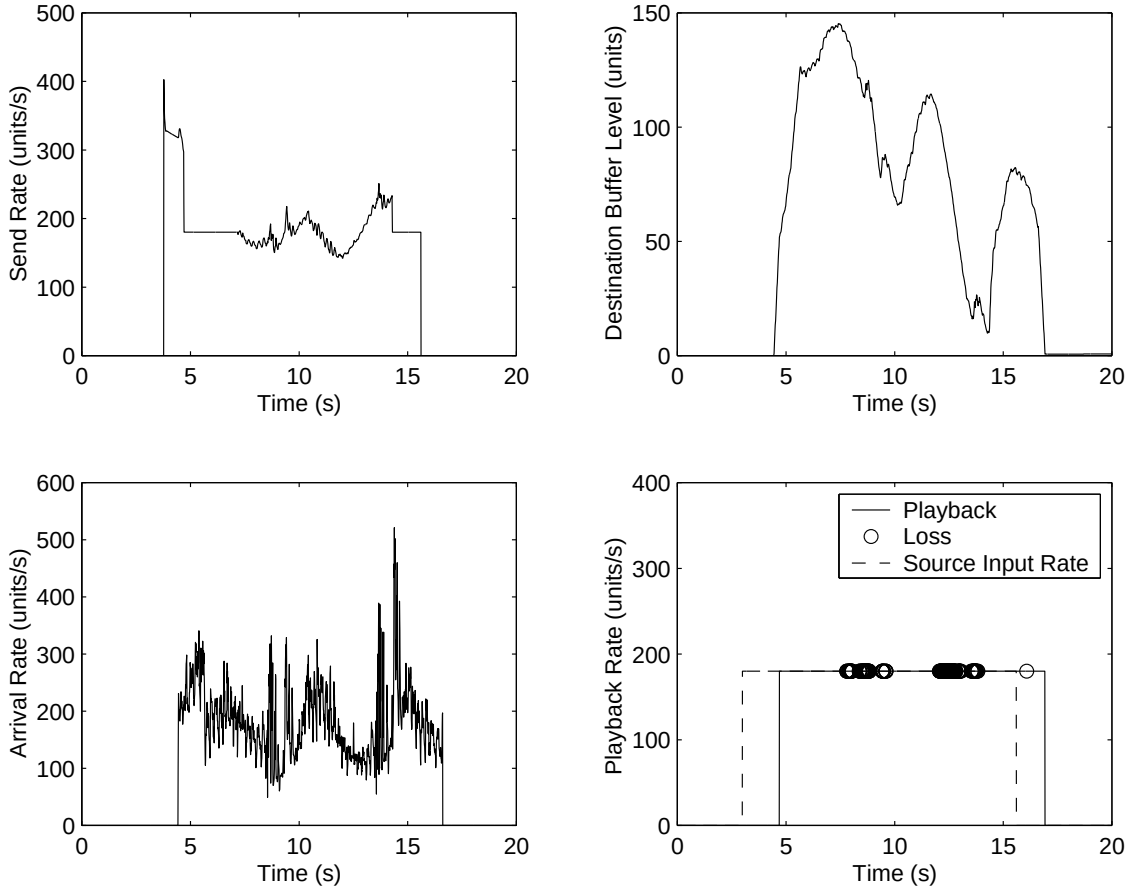

Fig. 85. Buffer Level and Flow Rates for Reactive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups.

4. Reactive Implementation of Controller 3

The system response for Controller 3 simulation is shown in Figures 85, 86 and 87. A 14% decrease on the losses is achieved. This occurs with the cost of 7% increase in dead-time.

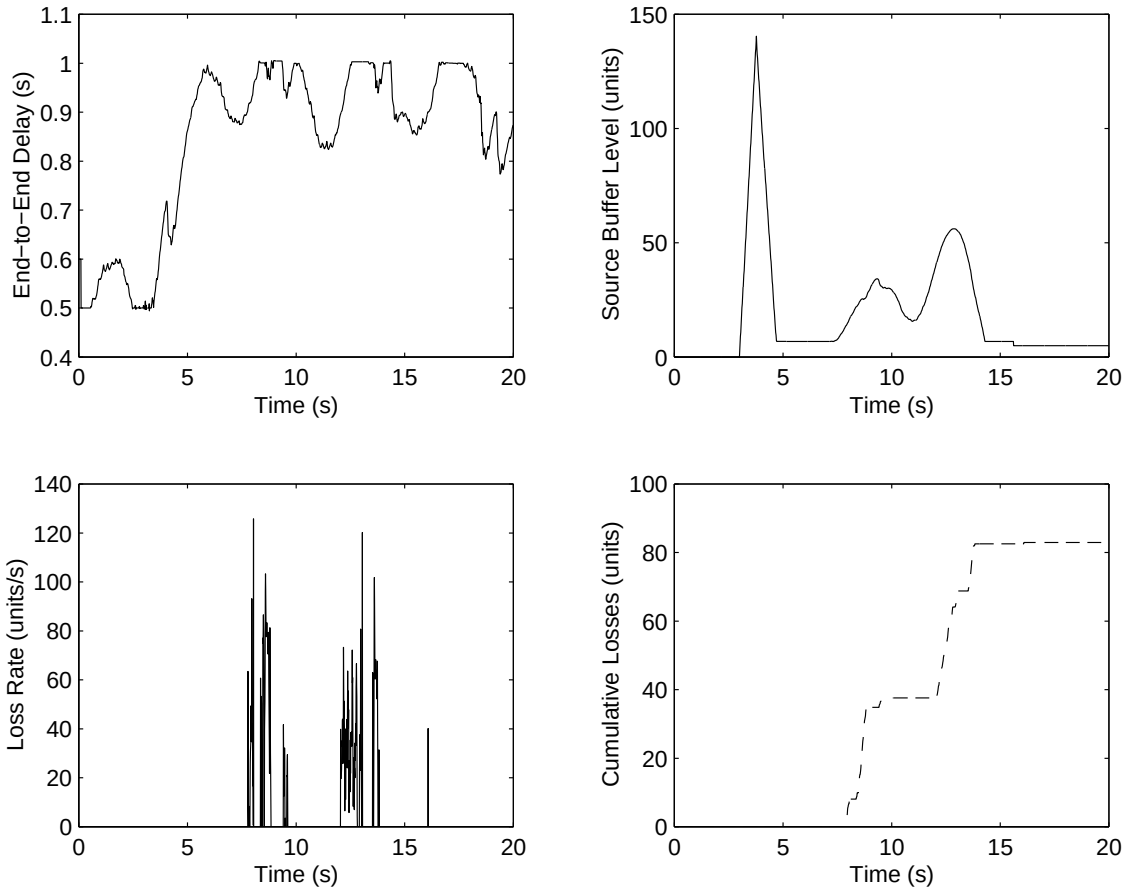

Fig. 86. End-to-End Delay, Source Buffer Level and Losses for Reactive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups.

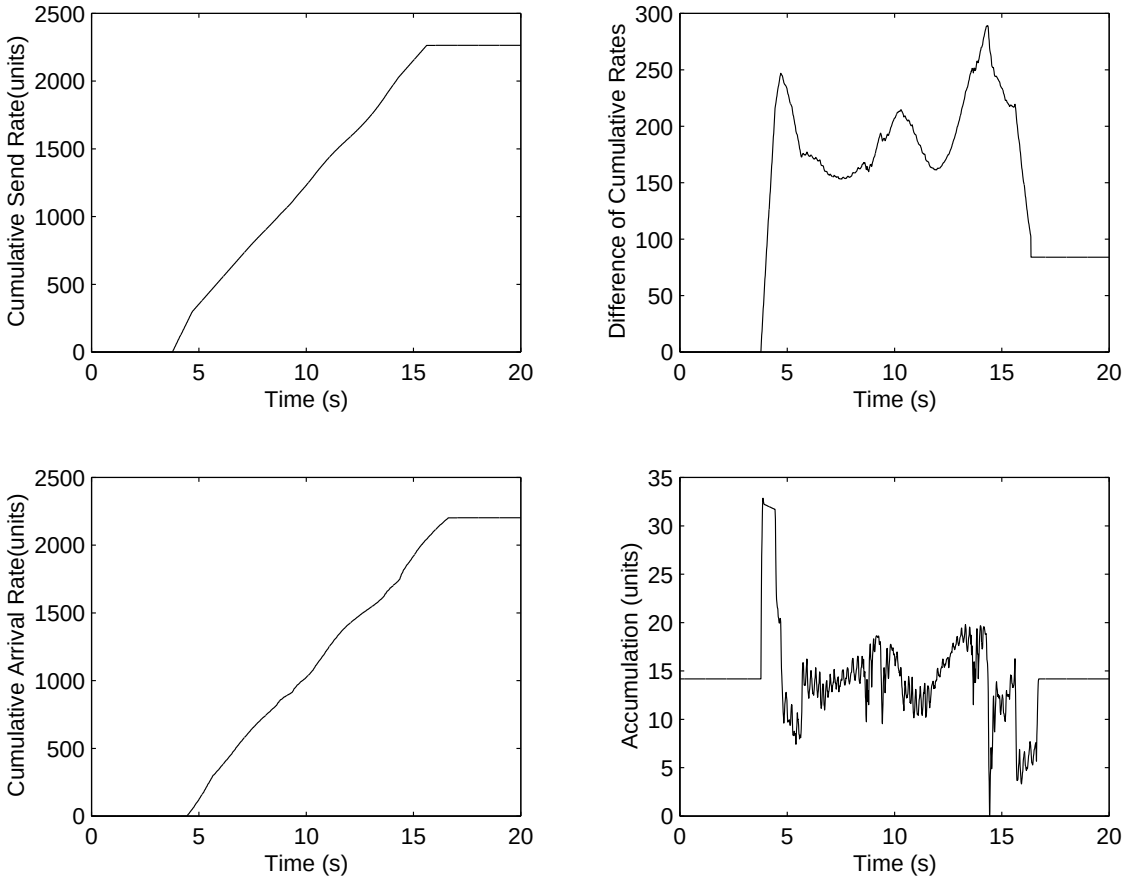

Fig. 87. Cumulative Flow Rates and Accumulation for Reactive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups.

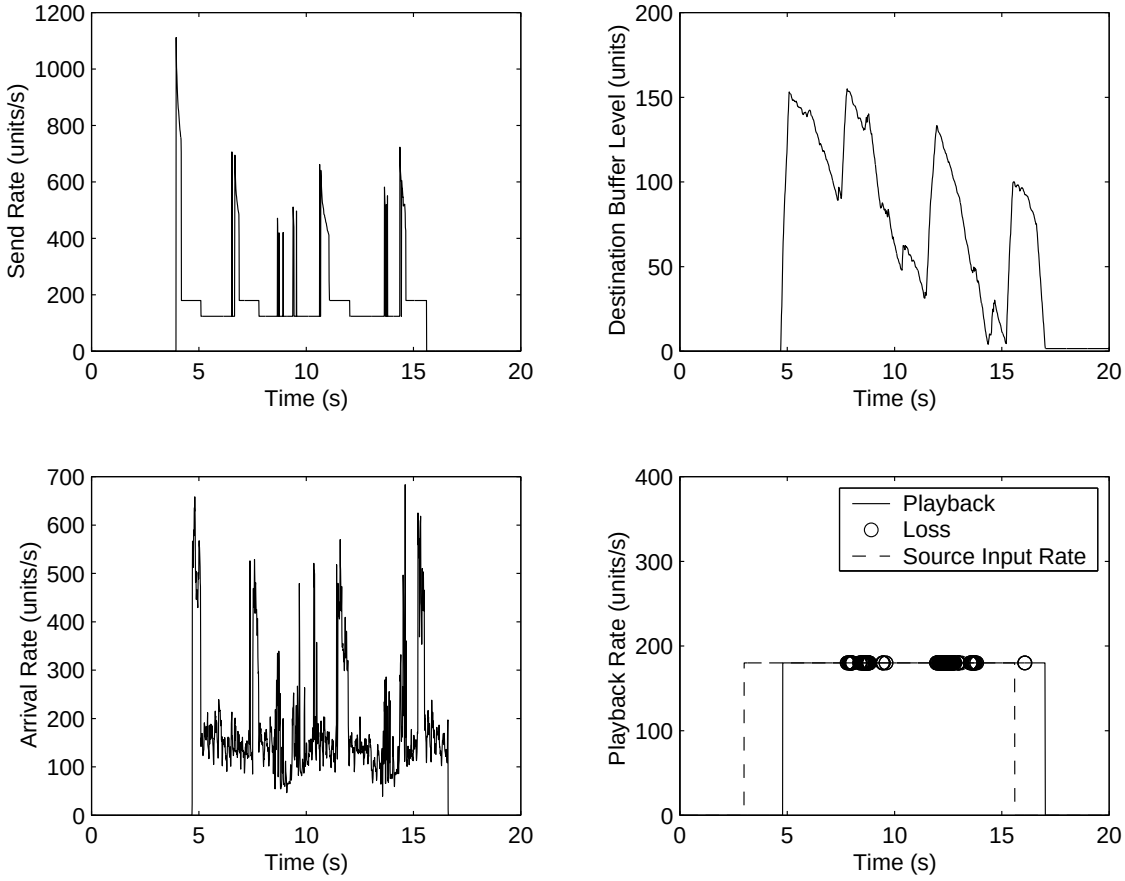

Fig. 88. Buffer Level and Flow Rates for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups.

5. Predictive Implementation of Controller 3

The system response for Controller 3 simulation is shown in Figures 88, 89 and 90 when it is implemented predictively. A 21% decrease on the losses is the result of the use of future information although dead-time is increased by 16%.

Figures 91 and 92 again illustrate the significance of the gain k_2. For $k_2 = 0.2$ the losses are reduced by 45%. However, the quality of the playback is poor because of multiple disruptions.

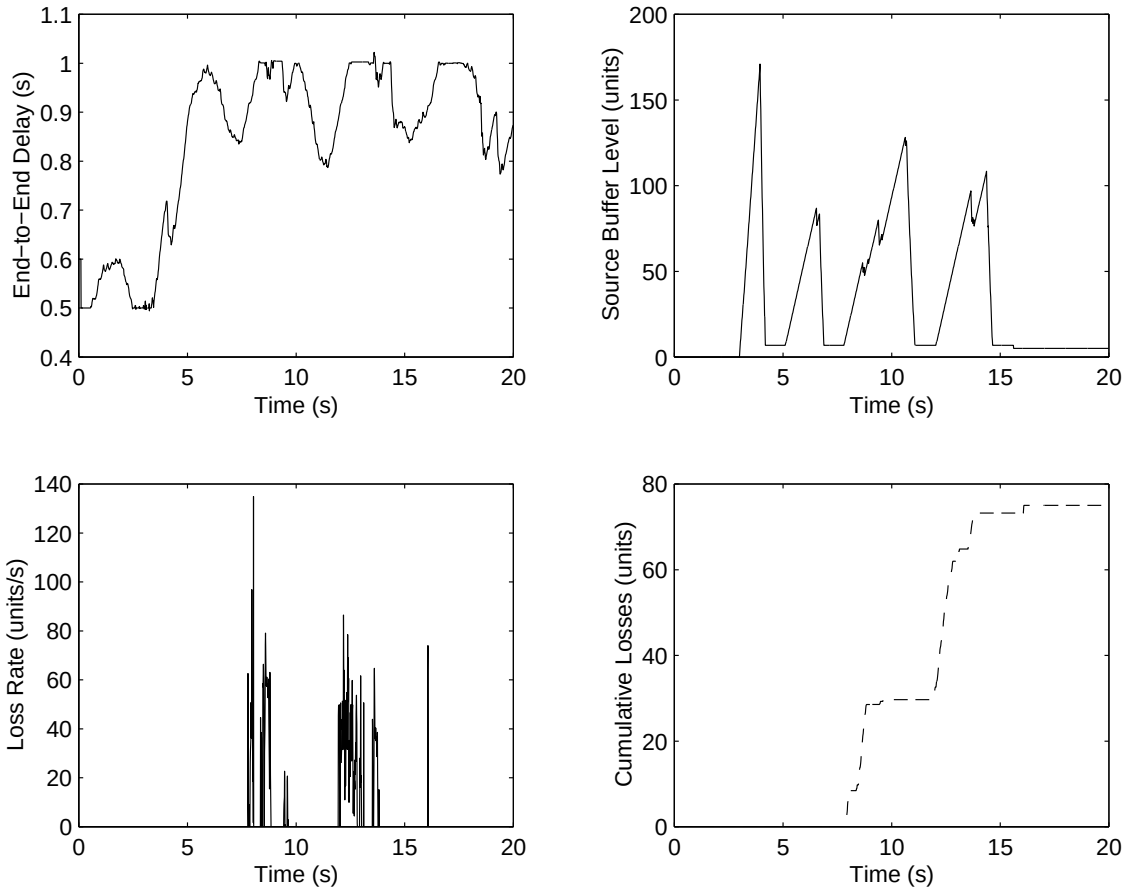

Fig. 89. End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups.

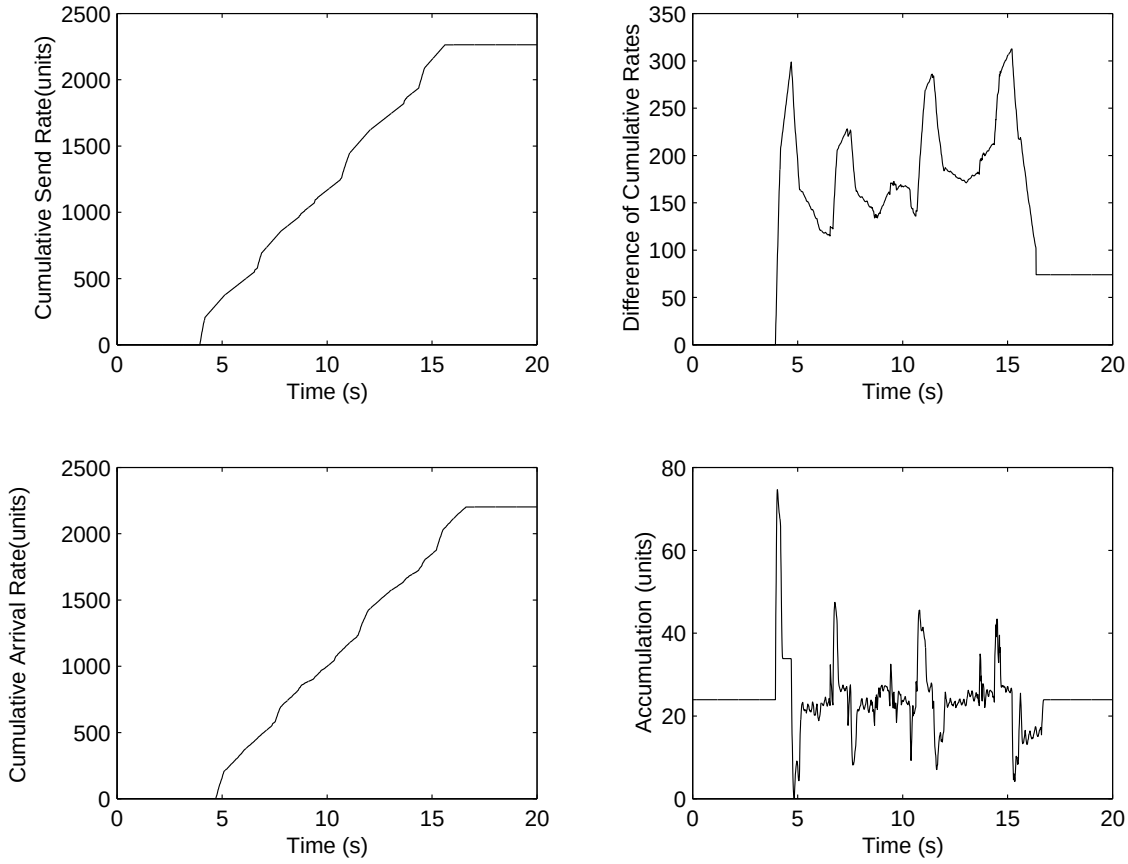

Fig. 90. Cumulative Flow Rates and Accumulation for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups.

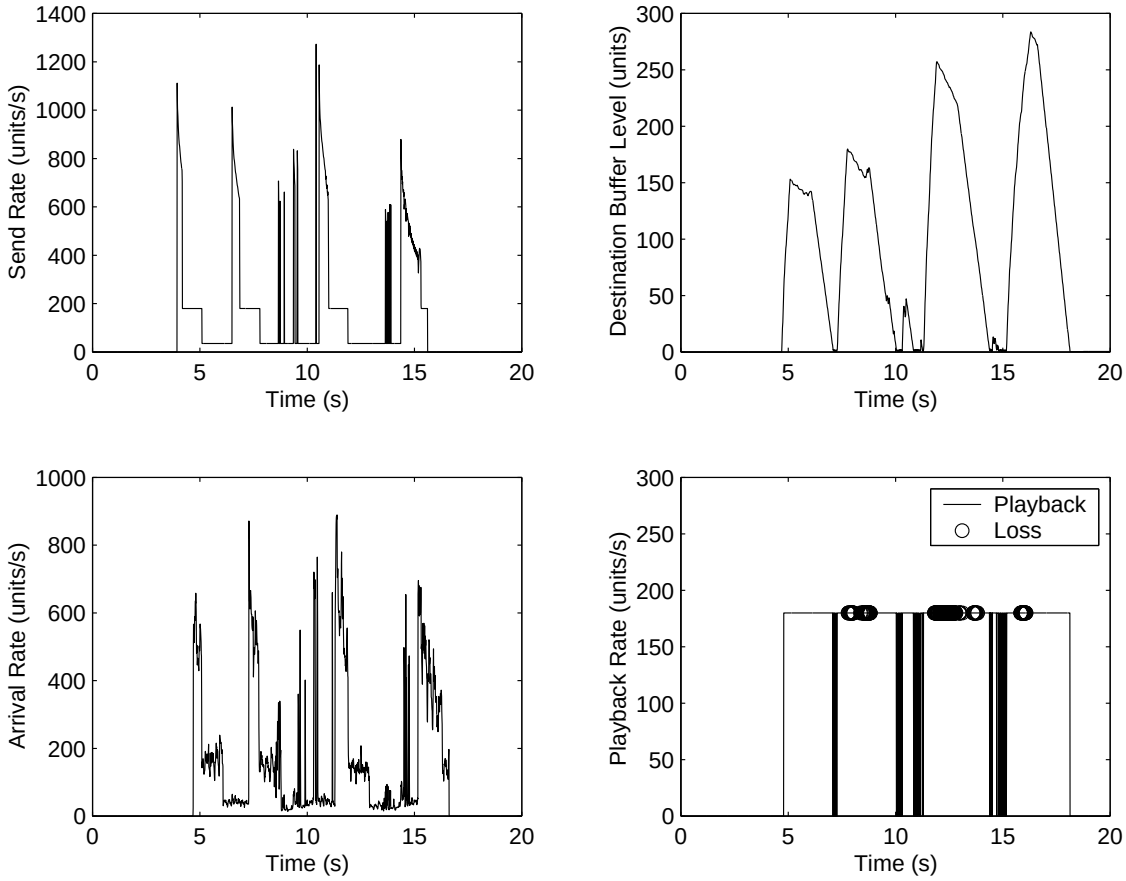

Fig. 91. Buffer Level and Flow Rates for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups and $k_2 = 0.2$.

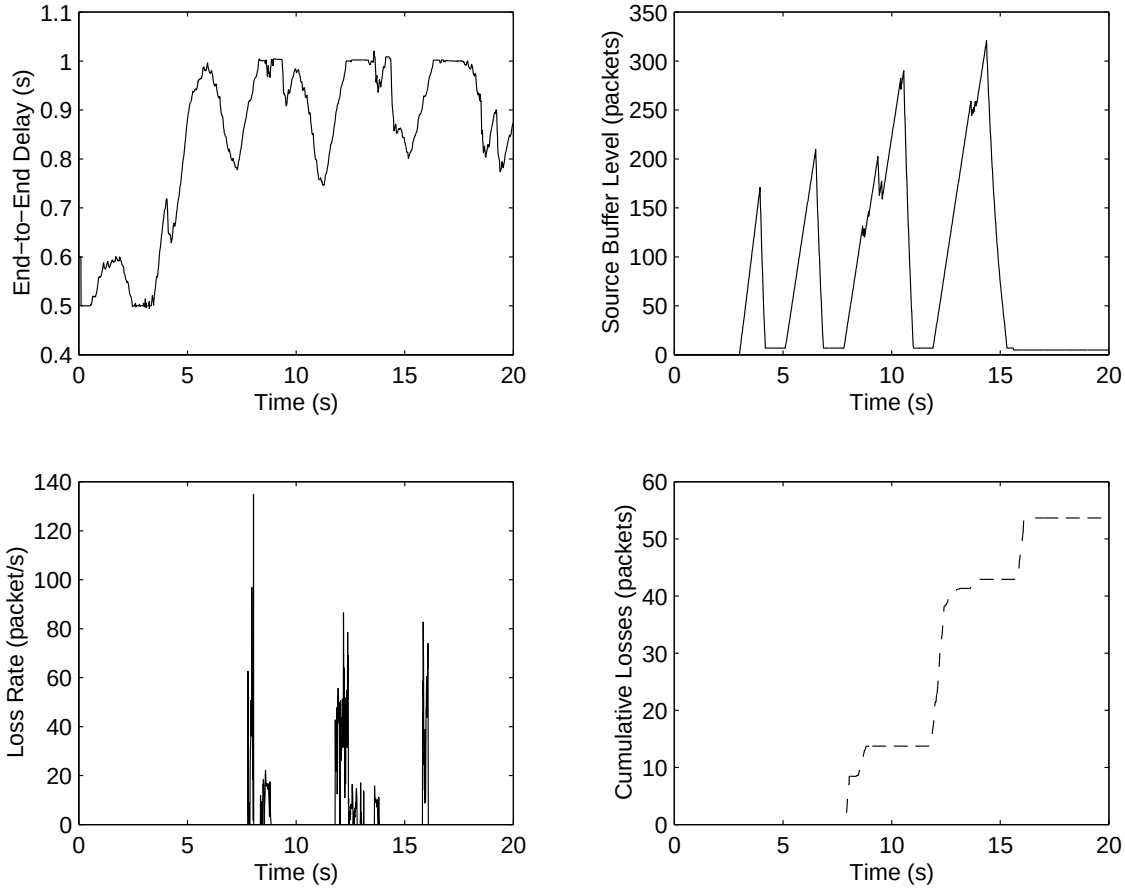

Fig. 92. End-to-End Delay, Source Buffer Level and Losses for Predictive Controller 3 Simulation Using Cross-Traffic 1 for Application Send Rate of 180 ups and $k_2 = 0.2$.

The controllers seem less effective in terms of loss reduction, for increased send rates. This is reasonable since the increased send rate is equivalent to a baseline send rate over a more congested network. The more the network is congested the less flexibility the controllers have.

On the other hand, the system dead-time seems to decrease when the send rate is higher. This is expectable since the higher the send rate is the less time is needed for the same amount of units to be stored in the source buffer.

E. Comparison of Controller Performance

A performance summary of the various controllers is given in the Tables III, IV, V and VI. All of the tables present the percentage change of the unit losses as well as the percentage of change in the system dead-time as compared to the uncontrolled case.

Table III, compares the controllers in the case where cross-traffic 1 is used. For cross-traffic 1 all of the non-linear controllers are able to reduce losses. However, Controller 2 is more effective than Controller 3.

Table IV shows the performance of the controllers when cross-traffic 2 is used. All non-linear control schemes are able of benefit in terms of unit losses.

Tables V and VI summarize the results of the simulations, when the application send rate is changed. In both cases the results are consistent with the results obtained for the baseline send rate. Nonetheless, the predictive schemes seem to have an advantage over the reactive control methods. The versatility of Controllers 2 and 3 is evident.

Each of the controllers has distinct advantages and disadvantages. These are summarized in Table VII where open-loop, linear, reactive, and predictive control

Table III. A Comparison of Controller Performance for Baseline Send Rate Using Cross-Traffic 1 Trace.

Method	**Percent Change in the Number of Losses Compared to Open-Loop**	**Percent Change in the Dead-Time Compared to Open-Loop**
Controller 1	+7%	+11%
Controller 2 Implemented Reactively	-22%	+13%
Controller 2 Implemented Predictively	-40%	+18%
Controller 3 Implemented Reactively	-16%	+20%
Controller 3 Implemented Predictively	-33%	+14%
Model Predictive Control (Destination Buffer Regulation)	0%	+18%
Model Predictive Control (Cumulative Flow Difference Regulation)	0%	+9%

Table IV. A Comparison of Controller Performance for Baseline Send Rate Using Cross-Traffic 2 Trace.

Method	**Percent Change in the Number of Losses Compared to Open-Loop**	**Percent Change in the Dead-Time Compared to Open-Loop**
Controller 1	+15%	+10%
Controller 2 Implemented Reactively	-9%	+14%
Controller 2 Implemented Predictively	-19%	+17%
Controller 3 Implemented Reactively	-14%	+20%
Controller 3 Implemented Predictively	-37%	+32%
Model Predictive Control (Destination Buffer Regulation)	0%	+10%

Table V. A Comparison of Controller Performance for 120 ups Send Rate Using Cross-Traffic 1 Trace.

Method	**Percent Change in the Number of Losses Compared to Open-Loop**	**Percent Change in the Dead-Time Compared to Open-Loop**
Controller 2 Implemented Reactively	-33%	+20%
Controller 2 Implemented Predictively	-45%	+23%
Controller 3 Implemented Reactively	-20%	+17%
Controller 3 Implemented Predictively	-40%	+23%

Table VI. A Comparison of Controller Performance for 180 ups Send Rate Using Cross-Traffic 1 Trace.

Method	**Percent Change in the Number of Losses Compared to Open-Loop**	**Percent Change in the Dead-Time Compared to Open-Loop**
Controller 2 Implemented Reactively	-13%	+3%
Controller 2 Implemented Predictively	-20%	+17%
Controller 3 Implemented Reactively	-14%	+7%
Controller 3 Implemented Predictively	-21%	+16%

solutions are compared.

The open-loop case is the one used in most current real-time applications. The UDP use at the transport layer is suitable for all applications with strict delay constraints. It is the case by which the effectiveness of the feedback control options are determined.

The linear feedback control algorithms have poor QoS support because they do not consider the highly non-linear dynamics of best-effort networks. In most cases, linear feedback makes QoS and congestion worse, since it increases losses.

Reactive control laws are simple to implement because no predictor is required. When application controls are implemented reactively, QoS is improved in the case of simple networks where delays change slowly. However, the use of the same controllers in more "realistically" complicated networks might be questionable. Furthermore, Controller 2 uses delay measurements to determine the control effort. These measurements may not be available all of the times. In order for this controller to be implemented, it should be used in combination with devices of delay measurements. This issue is resolved by Controller 3. This controller has all the advantages without the need of delay measurements. However, as mentioned earlier the use of accumulation is not guaranteed to be reliable for all cases. The delay purely represents the network dynamics whereas the accumulation may be distorted by the send rate and sometimes may not clearly exhibit the actual network dynamics.

Predictive control laws are more complicated to implement than reactive control laws. However, they are more effective. In this research, the delay spikes are well defined and change slowly with respect to time. In more realistic network conditions, precise synchronization between controller and predictor would be needed. If the delay spikes have short duration, even small prediction errors can be detrimental. Nevertheless, for conditions similar to the conditions assumed in this work predictive

Table VII. An Overall Comparison of Controller Performance.

Method	**Advantages**	**Disadvantages**
Open-Loop	Simple On-Line Implementation	No QoS Guarantees
Linear Control	Simple On-Line Implementation	Poor QoS; Makes Congestion Worse
Non-Linear Controller 2 Implemented Reactively	Not much Computational Effort; Good QoS	Works Only for Specific Network Conditions (E2E Delay Changes Slowly with Time)
Non-Linear Controller 2 Implemented Predictively	Better QoS over the Reactively Implemented version	Requires Accurate End-to-End Delay Predictions; Difficult to Tune Control Parameters
Non-Linear Controller 3 Implemented Reactively	Simple Practical Implementation; Not much computational effort; Good QoS;	Difficult to Tune Control Parameters; Works Only for Specific Network Conditions (E2E Delay Changes Slowly with Time)
Non-Linear Controller 3 Implemented Predictively	Better QoS over the Reactively Implemented version	Requires Accurate End-to-End Delay Predictions; Difficult to Tune Control Parameters
Controller 4	Achieves its objective accurately	Complex On-Line Implementation; Much Computational Effort

control is proven to be the most effective solution.

In both reactive and predictive schemes, the controller gains must be tuned precisely to maintain a consistent send rate. Each controller term gain should be tuned, so that the specific term does not reduce the effects of the other terms of controller. If the integral control term gain is too high then the effect of the inverse delay or inverse accumulation control term is reduced. Tuning could be an area of future study. Adaptive tuning methods might be appropriate and should be investigated.

Special attention should be given to Controller 4. The MPC controller accurately tracks its objective. It's a systematic, adaptive algorithm applicable to a variety of systems with time-varying time-delays. In this work the use of a controller regulating one single objective was demonstrated. Since the objectives of the control problem discussed in this thesis are more than one, the use of a controller with predictors regulating multiple objectives should be considered. Controllers 2 and 3 which have multiple terms in their structure, each one responsible for a different objective. Similarly the MPC controller should attempt multiple objectives simultaneously. The assumption of the unconstrained control problem is also questionable. It is a simplifying but not realistic assumption. It is shown that the source buffer places constraints to the control effort. Therefore, including such constraints in the objective function (4.21) of the MPC controller should be considered. Implementation and study of such a technique could be a challenging but promising future area of research.

F. Chapter Summary

The benefits and drawbacks of application level feedback control are examined. Simple examples of each case are illustrated using MATLAB/Simulink in combination with files extracted from ns-2 software. The various compensation methods developed

are simulated and compared for their strengths and weaknesses.

CHAPTER VI

SUMMARY AND CONCLUSIONS

The objective of this research is to improve the QoS for real-time media applications over best-effort networks using unreliable transport protocols. This is achieved by considering several controller performance criteria and implementing several control schemes at the application level.

A. Summary

The problem is stated and objectives are set in Chapter I. Relevant literature is reviewed as an introduction to current research in this area.

The assumptions made in this research and issues regarding the problem considered are discussed in Chapter II. These issues are important to understanding the complications of the problem and how compensation may be achieved for real-time media over best-effort networks.

A continuous-time fluid-flow model of the system under investigation is developed in Chapter III. Systems of this type are very difficult to model accurately and therefore simplifying assumptions are made. This model is used for developing the control methods described in Chapters IV and V.

Several control methods are developed in Chapter IV. These control methods are evaluated in Chapter V using MATLAB/Simulink simulations combined with files extracted from ns-2. Methods for predictive flow control are presented in Chapter V. The main premise of the predictive controllers is that when the delay and congestion levels are high, the send rate is reduced to prevent losses and further network congestion.

The result is that QoS is impacted and improved compared to open-loop appli-

cation level control. The use of system identification for developing predictors and implementation of MPC control techniques is also demonstrated.

B. Conclusions

Some conclusions that can be made based on the results of this research include:

1. Predictive and even reactive control strategies can improve application level QoS in real-time applications compared to the open-loop case, when the conditions of the network are such that end-to-end delays change slowly as the delay spikes have long duration. In the case where delays change fast, reactive control can not be effective. The current congestion level information which is used by the reactive controller will be much different by the time the controller reacts.

2. In improving the real-time application QoS, reduction of losses is traded by increase in dead-time or playback disruptions or both. This is due to increased total end-to-end delay due to source buffering.

3. As the network congestion is decreased and losses are far apart and few, the benefit of application feedback control diminishes as there are no losses to prevent. The same can be said about networks that are very highly congested. Feedback control has little flexibility in congested networks where most flow is lost.

The last two are the key results of this research. It is shown that the playback disruptions can be eliminated by a larger buffer at the destination. Therefore the real benefit of the control techniques presented, is the reduction in the losses.

C. Recommendations for Future Work

This research shows that the QoS perceived by the end-user when delivering real-time media over best-effort networks can be improved using source-based predictive and reactive controllers. Application level end-to-end control strategies can be successfully used to prevent playback disruption and reduce losses in streamed media flows at the expense of stream dead-time. Some recommendations for future research include:

1. Modeling of networks with more realistic cross-flow traffic and multiple routers, should be attempted to explore the effectiveness of the developed controllers.

2. Better ns-2 support for end-to-end analysis and control needs to be developed. An interface between ns-2 and MATLAB needs to be considered.

3. The use of a MPC controller regulating multiple output signals should be considered.

4. Controller send rate constraints need to be accounted for by control algorithms. The use of constrained MPC algorithms should be considered.

5. Methods for the systematic tuning of application controller performance needs to be developed.

Special consideration needs to be given to how best to implement the controllers developed in this thesis. Also, the potential of the MPC algorithm should not be ignored. The use of a controller regulating two or more different objectives should be the next step although the computational complexity of this control solution, may require an unreasonable amount of computational resources. Instead, approximate controllers or other techniques may need to be considered.

This is a new area of research with great potential for growth and positive benefits to many areas. It requires extending control theory and developing new techniques. But above all, it requires new innovative ideas for more efficient use of network resources. And as the Internet develops, every contribution made for improving the QoS perceived by the end-users is significant.

REFERENCES

[1] J. Widmer, R. Denda, and M. Mauve, "A survey on TCP-friendly congestion control," *IEEE Network*, vol. 15, no. 3, pp. 28–37, 2001.

[2] P. Gevros, J. Crowcroft, P. Kirstein, and S. Bhatti, "Congestion control mechanisms and the best-effort service model," *IEEE Network*, vol. 15, no. 3, pp. 16–25, 2001.

[3] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, August 1999.

[4] S. Floyd, K. Fall, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *ACM SIGCOMM '00*, Stockholm, Sweden, August 2000, pp. 43–56.

[5] C. Barakat, "TCP/IP modeling and validation," *IEEE Network*, vol. 15, no. 3, pp. 38–47, May/June 2001.

[6] J. W. Roberts, "Traffic theory and the Internet," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 94–99, January 2001.

[7] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun, "On the nonstationarity of Internet traffic," in *Proceedings of ACM SIGMETRICS '01*, Cambridge, Massachusetts, June 2001, pp. 102–112.

[8] S. Floyd and V. Paxson, "Difficulties in simulating the Internet," *IEEE Transactions on Networking*, vol. 9, no. 4, pp. 392 – 403, August 2001.

[9] B. Liu, D. R. Figueiredo, Y. Guo, J. F. Kurose, and D. F. Towsley, "A study of networks simulation efficiency: Fluid simulation vs. packet-level simulation," in *INFOCOM*, 2001, pp. 1244–1253.

[10] V. Misra, W. B. Gong, and D. F. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *SIGCOMM*, 2000, pp. 151–160.

[11] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "A control theoretic analysis of RED," in *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, Anchorage, Alaska, USA, 2001, vol. 3, pp. 1510–1519.

[12] Y. Liu, F. L. Presti, V. Misra, D. Towsley, and Y. Gu, "Fluid models and solutions for large-scale IP networks," in *Proceedings of the 2003 ACM SIGMETRICS International Conference*, San Diego, California, USA, 2003, pp. 91 – 101.

[13] A. P. Black, J. Huang, R. Koster, J. Walpole, and C. Pu, "Infopipes: An abstraction for multimedia streaming," Technical Report Number CSE 02-001, Department of Computer Science and Engineering, OGI School of Science and Engineering, January 2002.

[14] S. Low, F. Paganini, and J. C. Doyle, "Internet congestion control," *IEEE Control Systems Magazine*, vol. 22, pp. 28–43, February 2002.

[15] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," in *Proceedings of SIGCOMM*, Pittsburgh, Pennsylvania, August 2002, pp. 537–549.

[16] O. C. Imer, S. Compans, T. Basar, and R. Srikant, "Available bit rate congestion control in ATM networks," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 38–56, February 2001.

[17] Y. Gu, H. O. Wang, Y. Hong, and L. G. Bushnell, "A predictive congestion control algorithm for high speed communications networks," in *Proceedings of the 2001 American Control Conference*, Arlington, Virginia, June 2001, vol. 5, pp. 3779–3780.

[18] Y. S. Sun, F. Tsou, M. Chang Chen, and Z. Tsai, "A TCP-friendly congestion control scheme for real-time packet video using prediction," in *Proceedings of the 1999 Global Telecommunications Conference*, Rio de Janeireo, Brazil, December 1999, vol. 3, pp. 1818–1822.

[19] J. Li and S. Kalyanaraman, "MCA: A rate-based end-to-end multicast congestion avoidance scheme," in *ICC 2002 - IEEE International Conference on Communications*, San Francisco, California, April 2002, pp. 2341–2346.

[20] Y. Xia, D. Harrison, S. Kalyanaraman, K. Ramachandran, and A. Venkatesan., "An accumulation-based congestion control model," in *IEEE International Conference on Communications*, Anchorage, Alaska, May 2003, pp. 657–663.

[21] J. Bolot, "Control mechanisms for packet audio in the Internet," in *Proceedings of the Fifteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, San Francisco, California, March 1996, vol. 1, pp. 232–239.

[22] S. Chakrabarti and R. Wang, "Adaptive control for packet video," in *Proceedings of the International Conference on Multimedia Computing and Systems*, Boston, Massachusetts, May 1994, pp. 56–62.

[23] T. Turletti and C. Huitema, "Videoconferencing on the Internet," *IEEE Transactions on Networking*, vol. 4, no. 3, pp. 340–350, June 1996.

[24] J. Bolot and T. Turletti, "A rate control mechanism for packet video in the Internet," in *Proceedings of the Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, Boston, Massachusetts, April 1994, vol. 3, pp. 1216–1223.

[25] S. C. Liew and D. C. Tse, "A control-theoretic approach to adapting VBR compressed video for transport over a CBR communications channel," *IEEE Transactions on Networking*, vol. 6, no. 1, pp. 42–55, February 1998.

[26] W. Stallings, *Data and Computer Communications*, Upper Saddle River, New Jersey: Prentice Hall, 1999.

[27] D. Minoli and A. Schmidt, *Internet Architectures*, New York: John Wiley & Sons, 1999.

[28] L. Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*, New York: McGraw-Hill, 2000.

[29] J. W. Mangan III, "Adaptive control of media applications in best effort networks using reliable transport protocols ," M.S. thesis, Texas A&M University, College Station, Texas, 2002.

[30] H. Ohsaki, M. Morita, and M. Murata, "On modeling round-trip dynamics of the Internet using system identification," *IEICE Transactions on Communication*, vol. E85-B, no. 1, pp. 1–9, January 2002.

[31] S. Mascolo, "Congestion control in high-speed communication networks," in *Automatica, Special Issue on Control Methods for Communication Networks*,

December 1999, pp. 1921–1935.

[32] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *Special Issue of IEEE Transactions on Automatic Control on Systems and Control Methods for Communication Networks*, vol. 47, pp. 945–959, 2002.

[33] C. V. Hollot, V. Misra, D. F. Towsley, and W. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *INFOCOM*, 2001, pp. 1726–1734.

[34] L. Wang and W. R. Cluett, "Tuning PID controllers for integrating processes," *IEEE Process Control Theory Applications*, vol. 144, no. 5, pp. 385–392, September 1997.

[35] S. Majhi and D. P. Atherton, "Modified Smith predictor and controller for processes with time delay," *IEEE Process Control Theory Applications*, vol. 146, no. 5, pp. 359–366, September 1999.

[36] P. Quet, B. Ataslar, A. Iftar, H. Ozbay, S. Kalyanaraman, and T. Kang, "Rate-based flow controllers for communication networks in the presence of uncertain time-varying multiple time-delays," *Automatica*, vol. 38, no. 6, pp. 917–928, March 2002.

[37] A. R. Osborn and D. W. Browning, "A comparative study of flow control methods in high-speed networks," in *Proceedings of the 12th Annual International Phoenix Conference on Computers and Communications*, Phoenix, Arizona, March 1993, pp. 353 – 359.

[38] P. H. Bauer, M. L. Sichitiu, and K. Premaratne, "Controlling an integrator

through data networks: Stability in the presence of unknown time-variant delays," in *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems*, Orlando, Florida, May 1999, vol. 5, pp. 491–494.

[39] P. H. Bauer, M. L. Sichitiu, and K. Premaratne, "Closing the loop through communication networks: The case of an integrator plant and multiple controllers," in *Proceedings of the 38th Conference on Decision and Control*, Phoenix, Arizona, December 1999, vol. 3, pp. 2180–2185.

[40] B. Ataslar, P. Quet, A. Iftar, H. Ozbay, T. Kang, and S. Kalyanaraman, "Robust rate-based flow controllers for high-speed networks: The case of uncertain time-varying multiple time-delays," in *Proceedings of the 2000 American Control Conference*, Chicago, Illinois, June 2000, vol. 4, pp. 2804–2808.

[41] G. A. Dumont, A. Elnaggar, and A. Elshafei, "Adaptive predictive control of systems with time-varying time delay," *International Journal of Adaptive Control and Signal Processing*, vol. 7, pp. 91–101, 1993.

[42] B. Li, D. Xu, and K. Nahrstedt, "Optimal state prediction for feedback-based QoS adaptations," in *Seventh International Workship on IWQoS*, London, England, June 1999, pp. 37–46.

[43] J. A. Rossiter, *Model-Based Predictive Control*, Boca Raton, Florida: CRC Press, 2000.

[44] D. W. Clarke and C. Mohtadi and P. S. Tuffs, "Generalized predictive control-Part I. The basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.

[45] D. W. Clarke and C. Mohtadi and P. S. Tuffs, "Generalized predictive control-Part II. Extensions and interpretations," *Automatica*, vol. 23, no. 2, pp. 149–160,

1987.

[46] R. Kvaternik, J. Juang, and R. Bennett, "Exploratory studies in generalized predictive control for active aeroelastic control of tiltrotor aircraft," in *NASA/TM-2000-210552*, October 2000.

[47] J. Juang and M. Phan, "Deadbeat predictive controllers," in *NASA Technical Memorandum, TM-112862*, May 1997.

[48] J. Juang and K. Eure, "Predictive feedback and feedforward control for systems with unknown disturbances," in *NASA/TM-1998-208744*, December 1998.

[49] "Network Simulator 2 (ns-2)," Information Science Institute, University of Southern California, 2002.

[50] S. Doddi, "Emperical modeling of end-to-end delay dynamics in best-effort networks ," M.S. thesis, Texas A&M University, College Station, Texas, 2003.

VITA

Apostolos Konstantinou was born in Thessaloniki, Greece. He received a Bachelor of Science degree in Mechanical Engineering from National Technical University of Athens in 2001. In May 2001, he joined the master's degree program in Mechanical Engineering at Texas A&M University.

Permanent Address:

1213 Holik Street,

Apt C,

College Station, TX 77840.

Contact email address: tolis_k@neo.tamu.edu, avk5473@hotmail.com

The typist for this thesis was Apostolos Konstantinou.